

Technology Center 2100

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

APPELLANT'S MAIN BRIEF ON APPEAL

APPLICANT:	Friedbert CRUSIUS	DOCKET NO:	P99,1248
SERIAL NO.:	09/341,207	ART UNIT:	2176
FILED:	July 7, 1999	EXAMINER:	A. Yuan (formerly A. Romero)
		Confirmation No.	4875

TITLE: METHOD FOR GENERATING A GROUP OF PAGE FILES
FORMATTED IN A PAGE MARKUP LANGUAGE

5 Mail Stop Appeal Brief-Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. §1.192, Appellant submits this Brief in support of the appeal of the above-referenced application, in triplicate, in support of the patentability of claims 1-10 finally rejected in the Office Action, dated July 2, 2003 ("OA"). A copy of the claims on appeal is attached as Appendix A, and a copy of the Final Office Action is attached as Appendix B. A Notice of Appeal was filed on October 31, 2003.

REAL PARTY IN INTEREST

20 The real party in interest in this appeal is the assignee, Siemens
Aktiengesellschaft, a German corporation.

RELATED APPEALS AND INTERFERENCES

There are no related appeals and no related interferences known to

Appellant, Appellant's Assignee, or Appellant's legal representative.

STATUS OF CLAIMS

Claims 1-10 are on appeal, and constitute all pending claims of the application. Claims 1-10 were rejected under 35 U.S.C. §102(e) as being
5 anticipated by Truong, U.S. Patent No. 6,151,609, which was the sole basis for rejection in the application. A copy of Truong is attached as Amendment C.

STATUS OF AMENDMENTS

Amendment B was filed on January 22, 2003. Due to non-receipt of this amendment by the Examiner, it was retransmitted to the Examiner on April 14,
10 2003 and entered by the Examiner. Thus, all amendments made in this case are of record and have been considered by the Examiner.

SUMMARY OF THE INVENTION

In the description below, the use of reference characters in the description is intended solely to assist in the understanding of the invention and is not
15 intended to be limiting in any sense unless clearly and explicitly indicated as such—otherwise, the reference characters should be considered as exemplary.

In general terms, the present invention is related to a method for providing information/documentation related to system processes and present this information in a simplistic way (page 1, paragraph beginning on line 6) (1/6). The
20 invention permits utilizing existing documentation as a basis (1/20). New documents (recovered data) are formed from old ones (original data) and shared via a network (1/29); this permits maximum access to this information while minimizing the storage required (2/1). Furthermore, the original data is protected

so that a user does not overwrite it (2/9)—this decouples the recovered data from the original data 2/18.

In order to accomplish this, an author file (AT) is created (drawn up – see 1/21) that has a data record structure (DSx or, the respective address reference 5 to these records, ATDSx) on an authoring system (AUTS) (claim 1(a.1), Figure 3) that is connectable to a server via a data transmission line (DL) (6/3).

This author file (AT) permits the editing of text and graphical information within a data record of the file (e.g., TGDx, Fig. 2), and permits the insertion of reference information (e.g., DS-ADR) about data records (DSx) of addressable 10 data record-structured files (DSDx) (nomenclature also DSDx:DSx for file and record within the file, e.g., DSD1:003 to reference the third record in the first file) in the authoring system (AUTS) (claim 1(a.2), Figure 3, pp. 6-7)). Note that the structure of the author file (AT) and the addressable data record-structured files (DSDx) is the same and thus they can be treated the same (6/3). The author file 15 (AT) is later stored as a new data record-structured file (DSDx+1) (8/1).

A page file (SDx) in a page markup language (e.g., HTML) is generated by a format generator (HTML-GEN) from both records (DSx) of the record-structured author file (AT) and data records belonging to the data record-structured files (DSDx:DSx) (claim 1(b), Figure 3, 7/24).

20 A page markup language-specific link control address (e.g., HTML-ADR <XC> in Figure 3) addressing a page file (SDx) associated with the marked data record containing reference information inserted in the data records (DSx) (claim 1(c), 7/5-14, bottom portion of Figure 3). This link control address (HTML-ADR)

preamble of claim 1 at 1/31-50 and 5/7-53, teaching a group of web pages (present invention(PI): page files) in HTML format stored in a server to be accessed by a plurality of users.

The Examiner indicated that the claim element 1(a.1) is disclosed by
5 Truong 3/11-42, 10/15-35 and 11/35-57, teaching an editing system that allows a file to be edited that can be implemented on any computer or interconnected computer, and that claim element 1(a.2) is disclosed in this section of Truong by its teaching that the file defines a web page that can be accessed with its filename (PI: reference information), and associated added URL (PI: address).

10 The Examiner further indicated that claim elements 1(b)-(d) are taught by Truong at 3/11-43, 7-20-28, and 10/15-35 by its teaching of storing a web page formatted in HTML and retrieving it from a file defining the web page where the file is referred to by a filename (PI: reference information) and identified and associated with a URL (PI: address). The Examiner indicated that claim element
15 1(e) is disclosed by Truong at 5/54-61, 7/20-28, and 8/38-53 by its teaching that web pages with a URL or address is sent to the server.

The Examiner indicated that claim 2 is taught by Truong at 5/54-61, 7/20-28, and 10/15-35 by teaching a URL or address of a web page defined from a file where the file is referred to with a filename (PI: reference information) to be
20 viewed and edited. The Examiner indicated that claim 3 is taught by Truong at 3/26-47 and 4/53-61 by its teaching that filenames (PI: reference information) of files can be added and stored. The Examiner indicated that claim 4 is taught by Truong at 7/20-28 by its teaching that a file defining a web page with a URL is

located at a server. The Examiner indicated that claim 5 is taught by Truong at 3/11-43 by its teaching of an identification of files with filenames (PI: reference information) to be viewed and edited and at 11/9-19 by its teaching that editing consists of inserting or adding to the file.

5 The Examiner indicated that claim 6 is taught by Truong at 2/17-31, 6/55-67, and 9/20-26 by its teaching of fields or sections within a web page with an assigned URL (PI: address) and at 7/20-28 teaches a web page defined within a file, wherein the file is referred to with a filename (PI: reference information) to be selected for viewing and editing. The Examiner indicated that claim 7 is taught by
10 Truong at 1/31-67 and 10-15-35 by its teaching of selecting a stored file at the server, wherein the file defining a web page contains an assigned URL (PI: address). The Examiner indicated that claim 8 is taught by Truong at 1/31-50 and 5/54-61 by its teaching of a URL (PI: address) of a web page. The Examiner indicated that claim 9 is taught by Truong at 5/54-61, 7/20-28, and 8/38-53 by its
15 teaching that web pages addressed with a URL (PI: address) are located at a server for retrieval and the file defining the web page is to be viewed and edited). The Examiner indicated that claim 10 is taught by Truong at 1/51-67 and 5/54-61 by its teaching of using a web browser for navigating web pages displayed to the user on a computer system.

20 In responding to arguments made in Amendment B, the Examiner indicated that Truong discloses, at 3/11-18, a remote editor system that allows files stored on an Internet server to be remotely edited, where the files are selected and edited or viewed by a client using a browser (i.e., the file is edited (authored) and formatted (converted) to be displayed to the user.

The Examiner further indicated Truong discloses a “format generator” at 7/20-28 by its teaching that a web browser receives information in HTML format to be interpreted and displayed to the client, and, at 6/55-67, that a web browser formats the document directed by HTML tags to be displayed by the user.

5 The Examiner indicated that Truong discloses a “page markup language specific link control address” at 10/15-35 by teaching viewing or editing a desired file corresponding to a filename, where the filename is sent to the server to retrieve the desired stored file and determining whether the file contains HTML tags to be interpreted (formatted) by the web browse; also, at 5/54-67 teaching
10 providing the address or URL of a web page, where the web page is generally stored on a server to be retrieved for viewing by the client.

The Examiner also stated that Truong discloses (with respect to claim 3), at 3/26-43, “reference information about other data records of the author file” by its disclosure of filenames identifying files included in the server path to be
15 retrieved for viewing, and furthermore, at 2/17-31, that a user can select a graphical icon or link on a displayed web page that will automatically take them to a desired web page.

The Examiner further stated that Truong discloses (with respect to claim 6), at 2/17-31, 6-55-67 and 9/20-26) a structured HTML web page with hypertext
20 links, where each link selected by the user will automatically take them to a desired web page, i.e., each hypertext link is assigned with a URL address identifying another web page to be retrieved for viewing.

The Examiner further stated that Truong discloses (with respect to claim

9), at 5/54-61, 7-20-28, and 8/38-53, selecting a radio button within a file selection displayed in HTML format for the retrieval of a desired file, where the server under control of a remote editor program communicates the text of the file to a web browser of the client for editing (change).

5 Finally, the Examiner stated that Truong discloses (with respect to claim 10), at 5/54-61, a client using a web browser to retrieve a desired web page, and furthermore at 2/17-31, a displayed web page in HTML format containing graphical icons or links for the user to conveniently navigate by clicking on a link to automatically take them to a desired web page.

10 ***Appellant's Position: Truong does not anticipate the present invention because it does not meet the all elements rule, i.e., it fails to teach each and every limitation of the claim language.***

 1. *Overview: The Truong reference discloses a system in which files stored on an Internet server can be edited with an Internet browser on a client,*
15 *whereas the present invention is directed toward the use of an authoring system that has a format generator for generating a group of page files from an author file as well as data records of other files in the authoring system that are identified by reference.*

 As described in more detail above, the present invention is directed to a
20 method for generating a group of page files formatted in a page markup language such as HTML. The page files are generated on the basis of an author file editable in an authoring system into which references to data records of other data record-structured files of the authoring system can be inserted. The author file is supplied to a format generator that generates a group of page files from the

author file as well as from the data records of the other files identified by the references.

Inventively, the references to the data records of the other files addressable in the authoring system are converted into markup language-specific

- 5 link control addresses (for example, HTML links) via which the pages files are linked on the server. In other words, the format generator of the present invention converts the reference structure established by the references of the files in the authoring system into a markup language-specific reference structure between the page files on the server that is realized via link control addresses.
- 10 The group of page files referencing one another that is generated by the format generator is subsequently transmitted to the server. Truong does not disclose such a mechanism.

- The Truong reference, in contrast, is directed to a “remote editor system” in which files stored on an Internet server can be edited with an Internet browser
- 15 running on a client. The underlying object in Truong thus fundamentally differs from the object of the subject matter of the present invention. Accordingly, the features of the respective technical solutions in Truong and in the subject matter of the application also have an entirely different inter-relationship.

2. *Truong fails to teach the format generator according to claim element*
- 20 *1(b) of the present invention, since it fails to teach sending an author file (having a record-structured format) to a format generator to produce a page file generated in a page markup language from the data records of the author file.*

Appellant does not believe that Truong teaches the format generator as

defined by claim element 1(b) of the present invention. As defined by this claim element, the format generator takes the author file, as defined by claim element 1(a) as an input, and creates a page file generated in a page markup language from the data records of the author file, as well as data records belonging to the
5 data record-structured files that are marked by reference information.

In the OA, p. 7, under numbered paragraph 10(A), the Examiner indicates that the present invention's "format generator" is taught by the web browser of Truong at 7/20-28 because its web browser receives information in HTML format to be interpreted and displayed to the client and at 6/55-67 because its web
10 browser formats the document directed by the HTML tags to be displayed to the user.

If the Examiner is equating the present invention's format generator with the web browser of Truong and is equating the author file with Truong's HTML file, then she fails to indicate how the web browser of Truong generates a page
15 file in page markup language from data records of the author file, as required by claim 1(b).

The only "generating" done by the web browser of Truong is to generate a transient display for the user. This comparison is improper for two reasons. The first is that the plain and ordinary meaning of the term "file" in the phrase "page
20 file" does not encompass a transient display on a computer screen, as would be necessitated by the Examiner's interpretation of this phrase. The second is that even if, for the sake of argument, the term "page file" could encompass such a definition, then such an interpretation is inconsistent with other elements of the

claim. For example, claim element 1(e) would require that the “page file”, i.e., transient display, be transmitted to the data server device via the data transmission line. Clearly this would make no sense, as the display of Truong (Truong’s web browser 32) is clearly associated with the client 12 (7/9-20, Figure 5 2), and not the server.

Furthermore, according to this interpretation, claim element 1(d) would require that the link control address be stored in the page file, i.e., the transient display generated from the HTML file. Clearly Truong does not effect the storage of a link control address on the transient display. Finally, according to claim 10 element 1(c), Truong would have to disclose generating a respective link control address addressing the page file, i.e., the transient display—it is difficult to envision what an address of a transient display would look like.

3. Truong does not disclose a page markup language-specific link control address that addresses a page file.

15 Truong reveals no teaching or suggestion for a format generator that generates a group of page files from the author file as well as from data records of other files of the authoring system that are identified by references. In particular, Truong yields no teaching of any kind to the effect that a page markup language-specific link control address that addresses a page file is respectively 20 generated from reference information inserted into the data records.

In the OA, p. 7, under numbered paragraph 10(B), the Examiner equates the link control address of the present invention with Truong’s filename, as taught at 10/15-35, that is selected by the user via a radio button and sent to the server

so that the file corresponding to the filename can be edited. Furthermore, in a telephone interview with the Examiner and her supervisor conducted on October 22, 2003, the Examiner indicated that the term “reference information” could be interpreted broadly enough to encompass the use of a “filename” as disclosed by

5 Truong.

Appellant respectfully disagrees, since this association ignores numerous interrelationships required by elements of claim 1. First, this link control address must address the page file associated with the marked data record (required by claim element 1(c)); the page file itself must be generated by the format

10 generator from the author file and is generated from data records of the author file (required by claim element 1(b)). Thus, there is a correspondence of the page files to individual records in the author file, and the link control address must address this page file (element 1(c))—furthermore, this link control address must be stored in the page file associated with the data record containing the reference

15 information (required by element 1(d))—the Examiner has failed to indicate how selecting a filename using radio buttons, sending the filename to a server requesting the file for editing, and editing this file relates to these claim elements of the present invention. In fact, there is no suggestion in Truong related to dealing with data records themselves—the portion of Truong’s disclosure cited by

20 the Examiner relates to the file level and not to any sort of breakout at the record level.

The identification of the reference information in the subject matter of the application with a file name in Truong is untenable since the two terms have a completely different interrelationship. Truong thus yields no teaching that

reference information (i.e., a filename) is inserted into a data record of the author file or of some other data record-structured file in order to generate a page markup language-specific link control address from this inserted file name using a format generator of the authoring system. Accordingly, Truong does not teach or
5 suggest that the generated link control address addresses that page file on the server that is allocated to the file referenced by the reference information (i.e. by the filename) on the authoring system.

Furthermore, the Examiner utilizes an inconsistent definition of the author file by interpreting it both as the network editor web page and as the server file to
10 be edited.

The HTML file mentioned in column 7, lines 20-28 in Truong should not be confused with the file to be edited that is mentioned in column 3, lines 11-43 and in column 10, lines 15-35. The HTML file displayed by the browser after the input of the URL, on the contrary, is the network editor web page with which a graphic
15 user interface is displayed for editing files. After the HTML file has been downloaded, the user must still define the file to be edited by clicking a radio button of the HTML file (9/6-16 and Figure 4).

In Truong, thus, it is not the file to be edited but the network editor web page that is identified by the input URL. The file to be edited is identified by its
20 file name on the server and transmitted from the server to the browser only after the selection of a radio button. A simultaneous interpretation of both the network editor web page and of the server file to be edited as author file is thus not consistent.

The files edited in the exemplary embodiments in Truong (see Figures 5 and 6 plus the appertaining text of the specification) are not HTML pages but, on the contrary, largely format-free shell scripts. The HTML keywords contained in the shell scripts are employed exclusively as parameters therein (for the 'echo' shell command). If the file to be edited were an HTML file, its HTML elements would interfere with the browser itself, and resulting in the file not being editable without further precautions (see column 10, lines 22-26 and Figure 3c, step 148).

Given these significant differences, Truong does not teach or suggest claim 1 of the present invention, nor any of the remaining dependent claims by virtue of their dependence.

4. Truong does not teach each and every element as required by claim 6 of the present invention.

Claim 6 of the present invention is directed to subdividing the information stored in the data record-structured files into information modules that are stored in a respective data record together with a respectively allocated structure address. For inserting a reference information referencing a data record of a reference file into a file, the structure addresses of the reference file are visualized in order to then select one of the structure addresses.

In the OA on p. 5, last carryover paragraph, the Examiner indicates that Truong at 2/17-31, 6/55-67 and 9/20-26 teaches fields or sections within a web page having an assigned URL (address), which the Examiner implicitly appears to be associated with data record-structured files that are subdivided into information modules.

Appellant can find no disclosure in Truong for structure addresses stored in a data record together with an information module for selection purposes along with the subdivision of record-structured files into information modules, as required by claim 6. Appellant does not understand what the reference file of

5 Truong would be equated to or how that, by visualization of the structure addresses filed in the reference file, one of the structure addresses would be selectable, and thus believes that Truong does not teach claim 6 of the present invention.

Therefore, Appellant respectfully contends that the present invention is not

10 anticipated by Truong.

CONCLUSION

For the above reasons, Appellants respectfully submits that the Examiner is in error in law and in fact in rejecting claims 1-10 based on the teachings of the above-discussed reference. Reversal of the rejection of all of those claims is
5 justified, and the same is respectfully requested.

This Brief is accompanied by a check in the amount of \$330.00, as required by 37 C.F.R. §1.17(c). If necessary, the Commissioner is hereby authorized to charge any additional fees which may be required to account No.
501519.

10

Respectfully submitted,

15

Mark Bergner (Reg. No. 45,877)
Mark Bergner
SCHIFF HARDIN & WAITE
Patent Department
6600 Sears Tower
233 South Wacker Drive
Chicago, Illinois 60606-6473
(312) 258-5779
Attorneys for Appellant
Customer No. 26574

20

CERTIFICATE OF MAILING

I hereby certify that an original and two copies of this correspondence are being
25 deposited with the United States Postal Service as First Class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, PO Box 1450, Alexandria, VA 22313-1450 on December 31, 2003

30

Mark Bergner

APPENDIX A CLAIMS INVOLVED IN THE APPEAL

***The bolded lettering on claim 1 is not a part of the claim, but has been
5 added to this appendix to assist in the discussion above.

1. A method for generating a group of page files formatted in a page
markup language for storage in a data server device of a data networking system
through which the page files are addressable by a multiplicity of data processing
10 user systems and are transferrable to the user systems, comprising the steps of:

[(a)] [(a.1)] drawing up a data record-structured author file on a data
processing authoring system which is connectable to the data
server device via a data transmission line, **[(a.2)]** in which author
file text and graphic information is editable within a respective data
15 record and reference information about data records of data
record-structured files which are addressable in the data processing
authoring system is insertable;

[(b)] sending the author file to a format generator device of the authoring
system, by which a respective page file is generated in the page
20 markup language from the data records of the author file and from
the data records which belong to the data record-structured files
and are marked by reference information;

[(c)] generating a respective page markup language-specific link control address addressing the page file associated with the marked data record from reference information inserted in the data records;

[(d)] storing the respective page markup language-specific link control address in the page file associated with the data record containing the reference information; and

[(e)] transmitting the generated page files that are provided with page markup language-specific link control addresses to the data server device via the data transmission line.

10

2. The method as claimed in claim 1, wherein reference information about other data records of data record-structured files which is addressable in the authoring system is addable to data records of the data record-structured files which are addressable in the authoring system.

15

3. The method as claimed in claim 1, wherein reference information about other data records of the data record-structured author file is addable to data records of the data record-structured author file.

20

4. The method as claimed in claim 1, wherein the author file is a data record-structured file which is already addressed in the authoring system.

5. The method as claimed in claim 1, wherein an item of reference information about files structured free of data records which is addressable in the authoring system is addable.

5 6. The method as claimed in claim 1, wherein the information stored in data record-structured files is subdivided into information modules to which at least one individual structure address is assigned, wherein each information module is stored together with its individual structure address in a respective data record, and wherein an item of reference information about a data record of a
10 reference file is added in that, by visualization of the structure addresses filed in the reference file, one of the structure addresses is selectable.

7. The method as claimed in claim 1, wherein when selecting a data record already stored in the data server device as a page file, a page markup
15 language-specific link control address addressing this page file is generated and is temporarily stored in a data field of the data record holding the reference information.

8. The method as claimed in claim 1, wherein the structure address is one
20 of an item of text information or an item of numerical information.

9. The method as claimed in claim 1, wherein a data record-structured file in the authoring system is addressable only if the page files assigned to data records thereof are already stored in the data server device, and wherein a page file is transmitted only if it is not yet stored or a change has been made to information content thereof, in particular of link control addresses.

10. The method as claimed in claim 1, wherein the stored page files are displayed in the user systems with navigation control fields which allow leafing through a group of page files to a logically next or preceding page file while avoiding activation of corresponding forward functions of a page access device.

**APPENDIX B
FINAL OFFICE ACTION**



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09:341,207	07/07/1999	FRIEDBERT CRUSIUS	P99.1248	4875

7590 07/02/2003

SCHIFF HARDIN & WAITE
Patent Department
6600 Sears Tower
Chicago, IL 60606-6473

EXAMINER

ROMERO, ALMARI DEL CARMEN

ART UNIT PAPER NUMBER

2176

DATE MAILED: 07/02/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

RECEIVED

JAN 07 2004

Technology Center 2100

RECEIVED

JUL 07 2003

SCHIFF HARDIN & WAITE
U.S. PATENT DEPT.

Office Action Summary

Application No.

09/341,207

Applicant(s)

CRUSIUS, FRIEDBERT

Examiner

Almari Romero

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 14 April 2003.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-10 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-10 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 04 April 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on _____ is: a) ☐ approved b) ☐ disapproved by the Examiner.
If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ 6) ☐ Other:

DETAILED ACTION

1. This action is responsive to communications: Amendment filed on 4/14/03.
2. The objection to the drawings has been withdrawn as necessitated by amendment.
3. The objection to the abstract of the disclosure has been withdrawn as necessitated by amendment.
4. The objection to the disclosure with regard to misspelled words has been withdrawn as necessitated by amendment.
5. The objection to claim with regard to defining an abbreviated term has been withdrawn as necessitated by amendment.
6. The objection the specification with regard to the reference "Russ et al." has been withdrawn based on Applicant's remarks on page 6, lines 6-14 in the amendment filed on 4/14/03.
7. Claims 1-10 are pending in the case. Claim 1 is independent claim.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) do not apply to the examination of this application as the application being examined was not (1) filed on or after November 29, 2000, or (2) voluntarily published under 35 U.S.C. 122(b). Therefore, this application is examined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

9. **Claims 1-10 are rejected under 35 U.S.C. 102(e) as being anticipated by Truong (USPN 6,151.609 – filed on 08/16/1996).**

Regarding independent claim 1, Truong discloses:

A method for generating a group of page files formatted in a page markup language, for storage in a data server device of a data networking system through which the page files are addressable by a multiplicity of data processing user systems and are transferable to the user systems (Truong on col. 1, lines 31-50 and col. 5, lines 7-53: teaches group of web pages (page file) in HTML format stored in server to be accessed by plurality of users by entering URL (address)), comprising the steps of:

(a) ^(a.1) drawing up a data record-structured author file up on a data processing authoring system which is connectable to the data server device via a transmission line, in which author file text and graphic information is editable within a respective data record and reference information about data records of data record-structured files which is addressable in the DP authoring system is addable (Truong on col. 3, lines 11-43, col. 10, lines 15-35, and col. 11, lines 35-57: ^(a.1) *Author File Setup* teaches editor system allows file to be edited which can be implemented on any computer or interconnected computer, ^(a.2) *AuthFile* wherein the file defines a web page which can be accessed with its filename (reference information) and associated added URL (address));

Art Unit: 2176

(b) sending the author file to a format generator device of the authoring system, by which a respective page file is generated in the page markup language from the data records of the author file and from the data records which belong to the data record-structured files and are marked by reference information, (c) a respective page markup language-specific link control address addressing the page file associated with the marked data record being generated from reference information added to the data records (d) and being stored in the page file associated with the data record containing the reference information (Truong on col. 3, lines 11-43, col. 7, lines 20-28, and col. 10, lines 15-35: teaches stored web page formatted in HTML and retrieved from a file which defines the web page and wherein the file is referred to by a filename (reference information) and identified and associated with a URL (address)).

(e) transmitting the generated page files that provided with page markup language-specific link control addresses, to the data server device via the data transmission line (Truong on col. 5, lines 54-61, col. 7, lines 20-28, and col. 8, lines 38-53: teaches web pages with URL or address is sent to server).

Regarding dependent claim 2, Truong discloses:

wherein reference information about other data records of data record-structured files which is addressable in the authoring system is addable to data records of the data record-structured files which are addressable in the authoring system (Truong on col. 5, lines 54-61, col. 7, lines 20-28, and col. 10, lines 15-35: teaches URL or address of web page defined from a file wherein the file is referred to with a filename (reference information) to be viewed and edited).

Regarding dependent claim 3, Truong discloses:

wherein reference information about other data records of the data record-structured author file is addable to data records of the data record-structured author file (Truong on col. 3, lines 26-47 and col. 4, lines 53-61: teaches filenames (reference information) of files can be added and stored).

Regarding dependent claim 4, Truong discloses:

wherein the author file is a data record-structured file which is already addressed in the authoring system (Truong on col. 7, lines 20-28: teaches file defining a web page with URL to be located at a server).

Regarding dependent claim 5, Truong discloses:

wherein an item of reference information about files structured free of data records which is addressable in the authoring system is addable (Truong on col. 3, lines 11-43: teaches identified files with filenames (reference information) to be viewed and edited and on col. 11, lines 9-19: teaches editing consists of inserting or adding to the file).

Regarding dependent claim 6, Truong discloses:

wherein the information stored in data record-structured files is subdivided into information modules to which at least one individual structure address is assigned, wherein each information module is stored together with its individual structure address in a respective data record, and wherein an item of reference information about a data record of a reference file is added in that, by visualization of the structure addresses filed in the reference file, one of the structure addresses is selectable (Truong on col. 2, lines 17-31, col. 6, lines 55-67, and col. 9, lines 20-26: teaches fields or sections within a web page with assigned URL (address) and on

col. 7, lines 20-28: teaches web page defined within a file, wherein the file is referred to with a filename (reference information) to be selected for viewing and editing).

Regarding dependent claim 7, Truong discloses:

wherein when selecting a data record already stored in the data server device as a page file, a page markup language-specific link control address addressing this page file is generated and is temporarily stored in a data field of the data record holding the reference information (Truong on col. 1, lines 31-67 and col. 10, lines 15-35: teaches selecting stored file at the server, wherein the file defining a web page which contains an assigned URL (address)).

Regarding dependent claim 8, Truong discloses:

wherein the structure address is one of an item of text information or an item of numerical information (Truong on col. 1, lines 31-50 and col. 5, lines 54-61: teaches URL (address) of a web page).

Regarding dependent claim 9, Truong discloses:

wherein a data record-structured file in the authoring system is addressable only if the page files assigned to data records thereof are already stored in the data server device, and wherein a page file is transmitted only if it is not yet stored or a change has been made to information content thereof, in particular of link control addresses (Truong on col. 5, lines 54-61, col. 7, lines 20-28, and col. 8, lines 38-53: teaches web pages addressed with URL (address) to be located at server for retrieval and file defining web page to be viewed and edited).

Regarding dependent claim 10, Truong discloses:

wherein the stored pages files are displayed in the user systems with navigation control fields which allow leafing through a group of page files to a logically next or preceding page file

while avoiding activation of corresponding forward functions of a page access device (Truong on col. 1, lines 51-67 and col. 5, lines 54-61: teaches web browser for navigating web pages displayed to the user computer system).

Response to Arguments

10. Applicant's arguments filed 4/14/03 have been fully considered but they are not persuasive.

A) Regarding Applicant's remarks on page 7, line 19 – page 8, line 5:

Truong discloses a remote editor system that allows files stored on an Internet server to be remotely edited; wherein files are to be selected and edited or viewed by a client using a browser (see col. 3, lines 11-18), in other words, the file is edited (authored) and formatted (converted) to be displayed to the user.

Truong discloses a “format generator” on col. 7, lines 20-28: teaches web browser receives information in HTML format to be interpreted and displayed to the client. Furthermore, Truong on col. 6, lines 55-67 discloses the web browser formats the document directed by the HTML tags to be displayed to the user.

B) Regarding Applicant's remarks on page 8, lines 3-5, lines 12-15, and page 10, lines 9-12:

Truong discloses “page markup language specific link control address”. Truong on col. 10, lines 15-35: teaches viewing or editing a desired file corresponding to a filename; wherein the filename is sent to the server to retrieve stored desired file and determining whether the file contains HTML tags to be interpreted (formatted) by the web browser. Furthermore, Truong on

col. 5, lines 54-67: teaches providing the address or URL of a web page; wherein the web page is generally stored on a server to be retrieving for viewing by client.

C) Regarding Applicant's remarks on page 9, lines 16-21:

Referring to claim 3, Truong does disclose "reference information about other data records of the author file". Truong on col. 3, lines 26-43 discloses filenames identifying files included in the server path to be retrieved for viewing. Furthermore, Truong on col. 2, lines 17-31: teaches the user can select a graphical icon or link on a displayed web page that will automatically take them to a desired web page.

D) Regarding Applicant's remarks on page 10, lines 6-8:

Referring to claim 6, Truong discloses a structured HTML web page with hypertext links; wherein each link selected by the user will automatically take them to a desired web page, in other words, each hypertext link is assigned with a URL address which identifies another web page to be retrieved for viewing (on col. 2, lines 17-31, col. 6, lines 55-67, and col. 9, lines 20-26).

E) Regarding Applicant's remarks on page 10, lines 13-18:

Referring to claim 9, Truong discloses selecting a radio button within a file selection displayed in HTML format (see col. 8, lines 34-37) for the retrieval of a desired file; wherein the server under control of remote editor program communicates the text of the file to web browser of client for editing (change) (on col. 5, lines 54-61, col. 7, lines 20-28, and col. 8, lines 38-53).

F) Regarding Applicant's remarks on page 10, lines 19-23:

Referring to claim 10, Truong discloses client using a web browser to retrieve desired web page (on col. 5, lines 54-61). Furthermore, Truong on col. 2, lines 17-31: teaches a

displayed web page in HTML format contains graphical icons or links for the user to conveniently navigate by clicking on a link to automatically take them to a desired web page.

Conclusion

11. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Art Unit: 2176

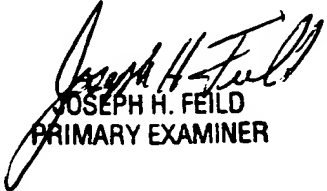
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Almari Romero whose telephone number is (703) 305-5945. The examiner can normally be reached on Mondays - Fridays (8:30am - 5:00pm).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Feild can be reached on (703) 305-9792. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7238 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-4700.

AR
June 29, 2003

EX FAX 703 746 9228


JOSEPH H. FEILD
PRIMARY EXAMINER

**REVISED AMENDMENT PRACTICE: 37 CFR 1.121 CHANGED
COMPLIANCE IS MANDATORY - Effective Date: July 30, 2003**

All amendments filed on or after the effective date noted above must comply with revised 37 CFR 1.121. See Final Rule: **Changes To Implement Electronic Maintenance of Official Patent Application Records** (68 Fed. Reg. 38611 (June 30, 2003)), posted on the Office's website at: <http://www.uspto.gov/web/patents/ifw/> with related information. The amendment practice set forth in revised 37 CFR 1.121, and described below, replaces the voluntary revised amendment format available to applicants since February 2003. **NOTE: STRICT COMPLIANCE WITH THE REVISED 37 CFR 1.121 IS REQUIRED AS OF THE EFFECTIVE DATE (July 30, 2003).** The Office will notify applicants of amendments that are not accepted because they do not comply with revised 37 CFR 1.121 via a Notice of Non-Compliant Amendment. See MPEP 714.03 (Rev. 1, Feb. 2003). The non-compliant section(s) will have to be corrected and the entire corrected section(s) resubmitted within a set period.

Bold underlined italic font has been used below to highlight the major differences between the revised 37 CFR 1.121 and the voluntary revised amendment format that applicants could use since February, 2003.

Note: The amendment practice for reissues and reexamination proceedings, except for drawings, has not changed.

REVISED AMENDMENT PRACTICE

I. Begin each section of an amendment document on a separate sheet:

Each section of an amendment document (e.g., Specification Amendments, Claim Amendments, Drawing Amendments, and Remarks) must begin on a separate sheet. Starting each separate section on a new page will facilitate the process of separately indexing and scanning each section of an amendment document for placement in an image file wrapper.

II. Two versions of amended part(s) no longer required:

37 CFR 1.121 has been revised to **no longer require two versions** (a clean version and a marked up version) of each replacement paragraph or section, or amended claim. Note, however, the requirements for a clean version and a marked up version for **substitute specifications** under 37 CFR 1.125 have been retained.

A) Amendments to the claims:

Each amendment document that includes a change to an existing claim, cancellation of a claim or submission of a new claim, **must include a complete listing** of all claims in the application. After each claim number in the listing, the status must be indicated in a parenthetical expression, and **the text of each pending claim** (with markings to show **current** changes) must be presented. The claims in the listing will replace all prior claims in the application.

- (1) The current status of all of the claims in the application, including any previously canceled, not entered or withdrawn claims, must be given in a parenthetical expression following the claim number using only one of the following seven status identifiers: (original), (currently amended), (canceled), (withdrawn), (new), **(previously presented) and (not entered)**. The text of all pending claims, **including withdrawn claims**, must be submitted each time any claim is amended. Canceled **and not entered** claims must be indicated by only the claim number and status, without presenting the text of the claims.
- (2) The text of all claims **being currently amended** must be presented in the claim listing with markings to indicate the changes that have been made relative to the immediate prior version. The changes in any amended claim must be shown by underlining (for added matter) or strikethrough (for deleted matter) with 2 exceptions: (1) for **deletion of five characters or fewer, double brackets may be used (e.g., [[eroor]]); and (2) if strikethrough cannot be easily perceived (e.g., deletion of the number "4" or certain punctuation marks), double brackets must be used (e.g., [[4]]).** As an alternative to using double brackets, however, **extra portions of text may be included before and after text being deleted, all in strikethrough, followed by including and underlining the extra text with the desired change (e.g., number 4 as number 14 as).** An accompanying clean version is not required and should not be presented. Only claims of the status "currently amended," and "withdrawn" that are being amended, may include markings.
- (3) The text of pending claims **not being currently amended, including withdrawn claims**, must be presented in the claim listing in clean version, i.e., without any markings. Any claim text presented in clean version will constitute an assertion that it has not been changed relative to the immediate prior version except to omit markings that may have been present in the immediate prior version of the claims.

- (4) A claim being canceled must be listed in the claim listing with the status identifier "canceled"; the text of the claim must not be presented. Providing an instruction to cancel is optional.
- (5) Any claims added by amendment must be presented in the claim listing with the status identifier "(new)"; the text of the claim must not be underlined.
- (6) All of the claims in the claim listing must be presented in ascending numerical order. Consecutive canceled, or not entered, claims may be aggregated into one statement (e.g., Claims 1 – 5 (canceled)).

Example of listing of claims (use of the word "claim" before the claim number is optional):

Claims 1-5 (canceled)

Claim 6 (previously presented): A bucket with a handle.

Claim 7 (withdrawn): A handle comprising an elongated wire.

Claim 8 (withdrawn): The handle of claim 7 further comprising a plastic grip.

Claim 9 (currently amended): A bucket with a ~~green~~ blue handle.

Claim 10 (original): The bucket of claim 9 wherein the handle is made of wood.

Claim 11 (canceled)

Claim 12 (not entered)

Claim 13 (new): A bucket with plastic sides and bottom.

B) Amendments to the specification:

Amendments to the specification, including the abstract, must be made by presenting a replacement paragraph or section or abstract marked up to show changes made relative to the immediate prior version. An accompanying clean version is not required and should not be presented. Newly added paragraphs or sections, including a new abstract (instead of a replacement abstract), must not be underlined. A replacement or new abstract must be submitted on a separate sheet, 37 CFR 1.72. If a substitute specification is being submitted to incorporate extensive amendments, both a clean version (which will be entered) and a marked up version must be submitted as per 37 CFR 1.125.

The changes in any replacement paragraph or section, or substitute specification must be shown by underlining (for added matter) or strikethrough (for deleted matter) with 2 exceptions: (1) for deletion of five characters or fewer, double brackets may be used (e.g., [leroor]); and (2) if strikethrough cannot be easily perceived (e.g., deletion of the number "4" or certain punctuation marks), double brackets must be used (e.g., [[4]]). As an alternative to using double brackets, however, extra portions of text may be included before and after text being deleted, all in strikethrough, followed by including and underlining the extra text with the desired change (e.g., number 14 as number 14 as)

C) Amendments to drawing figures:

Drawing changes must be made by presenting replacement figures which incorporate the desired changes and which comply with 37 CFR 1.84. An explanation of the changes made must be presented either in the drawing amendments, or remarks, section of the amendment. and may be accompanied by a marked-up copy of one or more of the figures being amended, with annotations. Any replacement drawing sheet must be identified in the top margin as "Replacement Sheet" and include all of the figures appearing on the immediate prior version of the sheet, even though only one figure may be amended. Any marked-up (annotated) copy showing changes must be labeled "Annotated Sheet Showing Changes" and accompany the replacement sheet as an appendix to the amendment. The figure or figure number of the amended drawing(s) must not be labeled as "amended." If the changes to the drawing figure(s) are not accepted by the examiner, applicant will be notified of any required corrective action in the next Office action. No further drawing submission will be required, unless applicant is notified.

Questions regarding the submission of amendments pursuant to the revised practice set forth in this flyer should be directed to: Elizabeth Dougherty or Gena Jones, Legal Advisors, or Joe Narcavage, Senior Special Projects Examiner, Office of Patent Legal Administration, by e-mail to patent.practice@uspto.gov or by phone at (703) 305-1616.

**APPENDIX C
RELEVANT REFERENCES**

United States Patent

Truong

[19]

[11] Patent Number: 6,151,609

[45] Date of Patent: *Nov. 21, 2000

[54] REMOTE EDITOR SYSTEM

[75] Inventor: Timothy M. Truong, Plano, Tex.

[73] Assignee: Electronic Data Systems Corporation, Plano, Tex.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: 08/699,076

[22] Filed: Aug. 16, 1996

[51] Int. Cl.⁷ G06F 17/12

[52] U.S. Cl. 707/505; 707/513; 709/219

[58] Field of Search 707/501, 505-508,
707/513, 530, 531; 709/201, 217, 218,
219; 345/335

[56] References Cited

U.S. PATENT DOCUMENTS

5,732,219	3/1998	Blumer et al.	395/200.57
5,793,966	8/1998	Amstein et al.	395/200.33
5,867,667	2/1999	Butman et al.	709/249
5,878,219	3/1999	Vance, Jr. et al.	709/217
5,898,835	4/1999	Truong	709/217

OTHER PUBLICATIONS

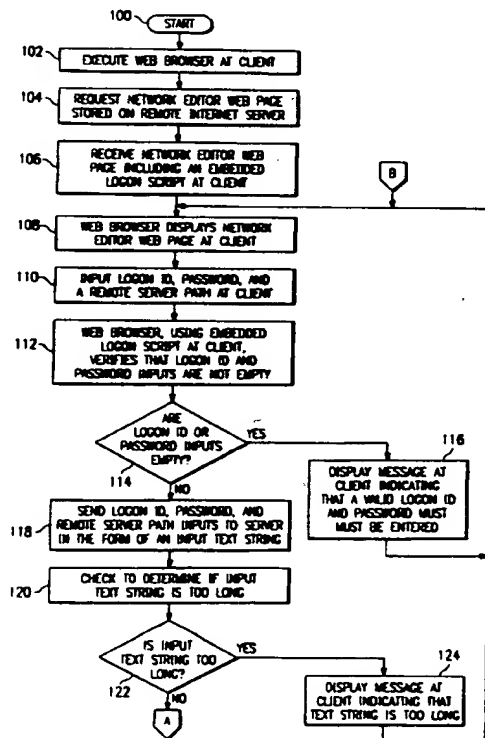
Microsoft, Windows 3.1 User's Guide, p. 64-65, 1992.
Laura Lemay, Teach Yourself Web Publishing with HTML in a Week, Sams Publishing, pp. 296-324, 1995.

Primary Examiner—Joseph H. Feild
Attorney, Agent, or Firm—L. Joy Griebenow; Baker & Botts L.L.P.

[57] ABSTRACT

A remote editor system (26) is provided for remotely editing files stored on a remote Internet server (15). The remote editor system (26) includes a client (12), using a web browser (32), and the remote Internet server (15) having a mass storage device (44), for storing computer files, and a processor (42). The processor (42) is responsive to one or more of the computer files of the mass storage device (44). The processor (42) is responsive to communicate an editor input form to the client (12) and to receive an input text string from the client (12) that includes a server path input that identifies a path of the remote Internet server (15). The processor (42) is further responsive to store the server path input as a variable and to communicate a file selection form to the client (12) so that a user may select a file to edit. The processor (42) receives a file selection input identifying the user's selection and provides the text of the selected file to the web browser (32) of the client (12) for editing. Once edited, the remote Internet server (15) receives the edited text and saves the file.

17 Claims, 7 Drawing Sheets



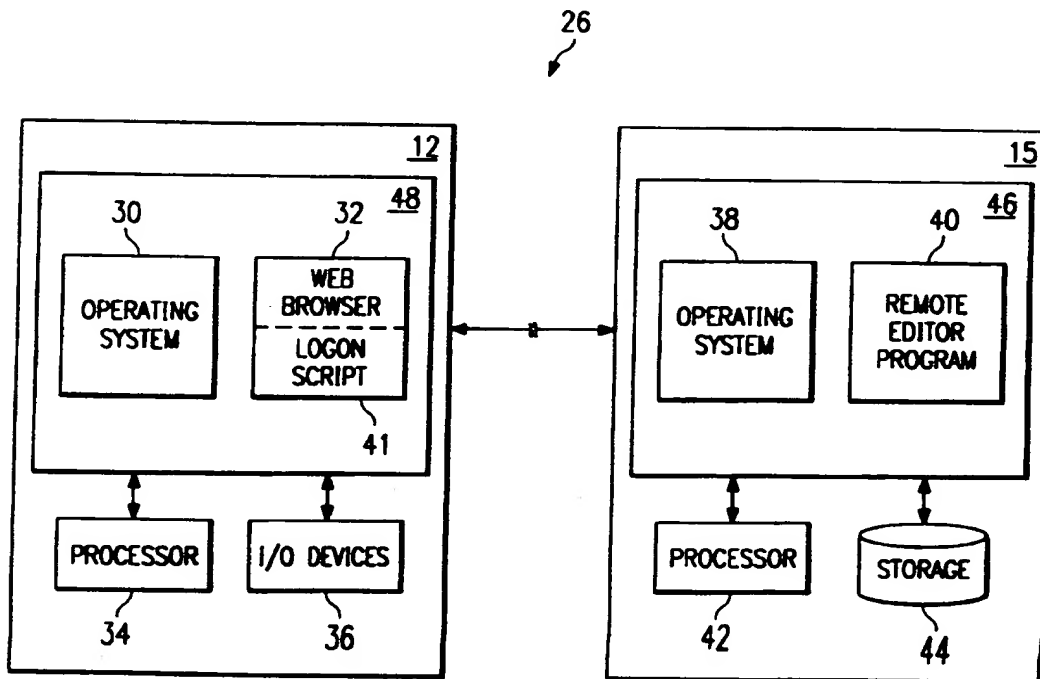
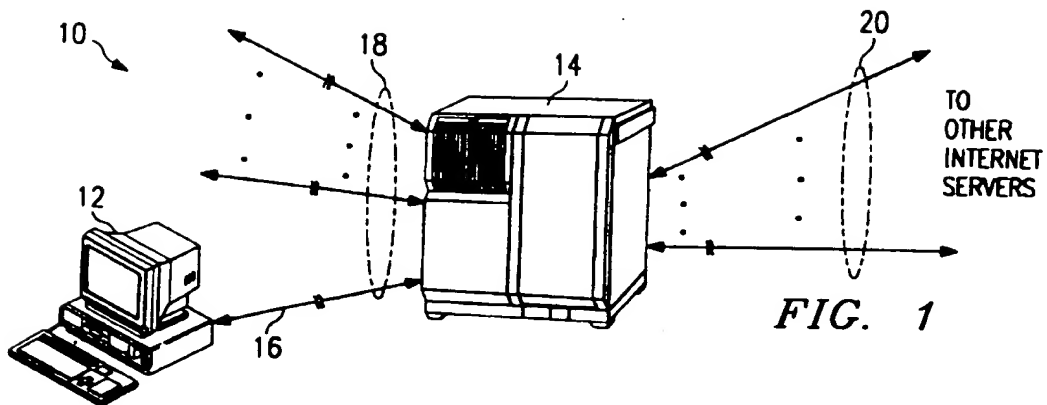
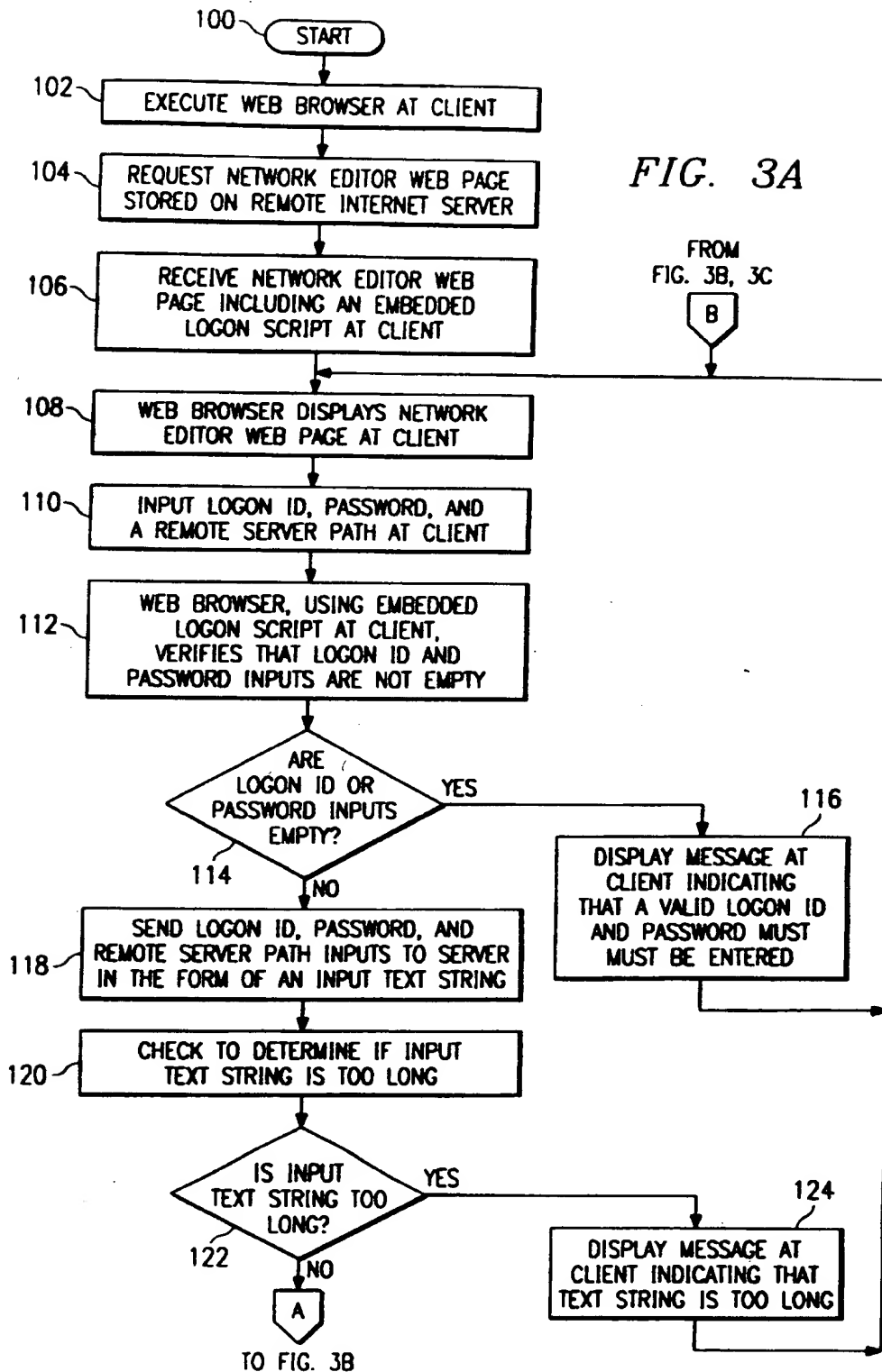


FIG. 2



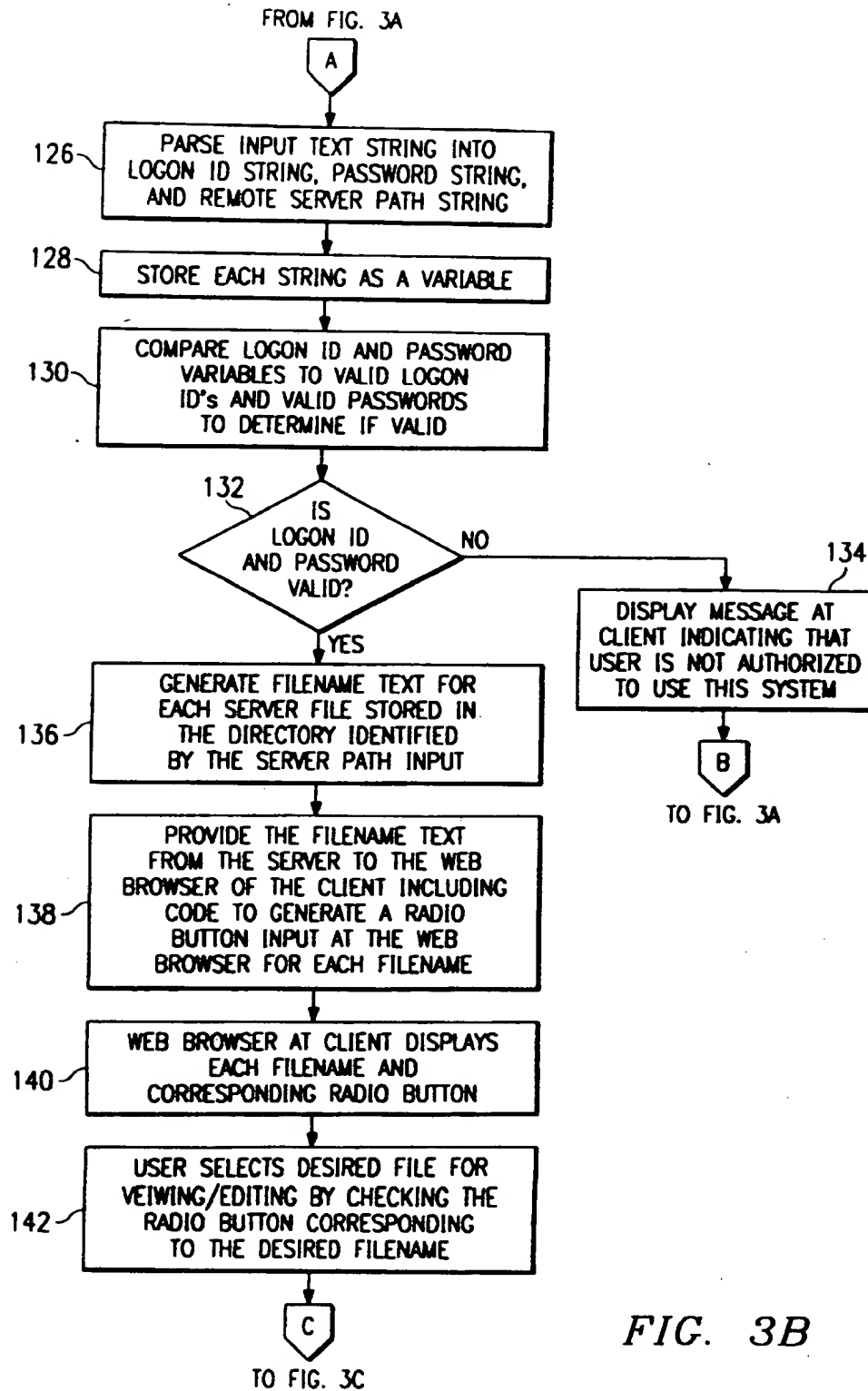
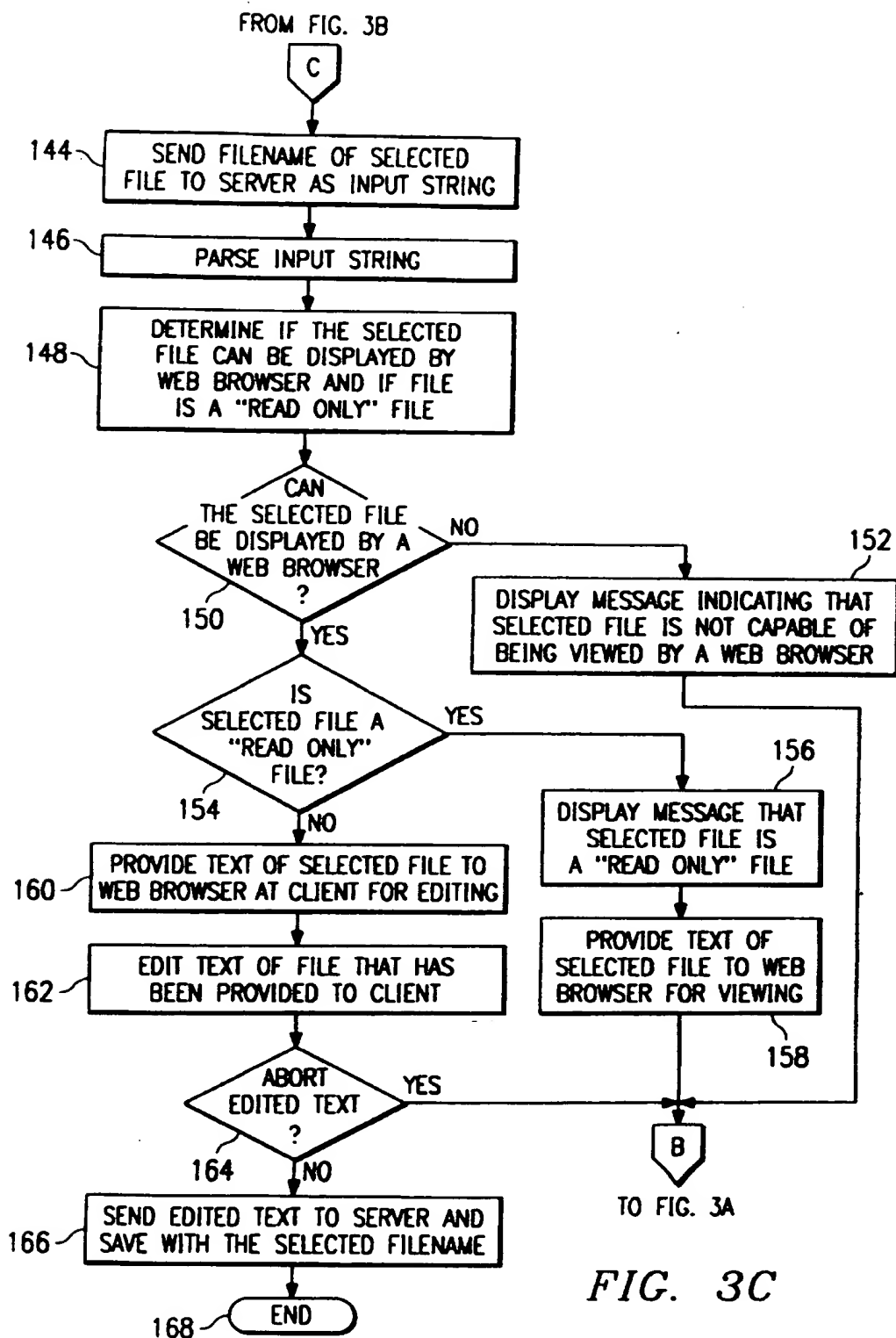


FIG. 3B



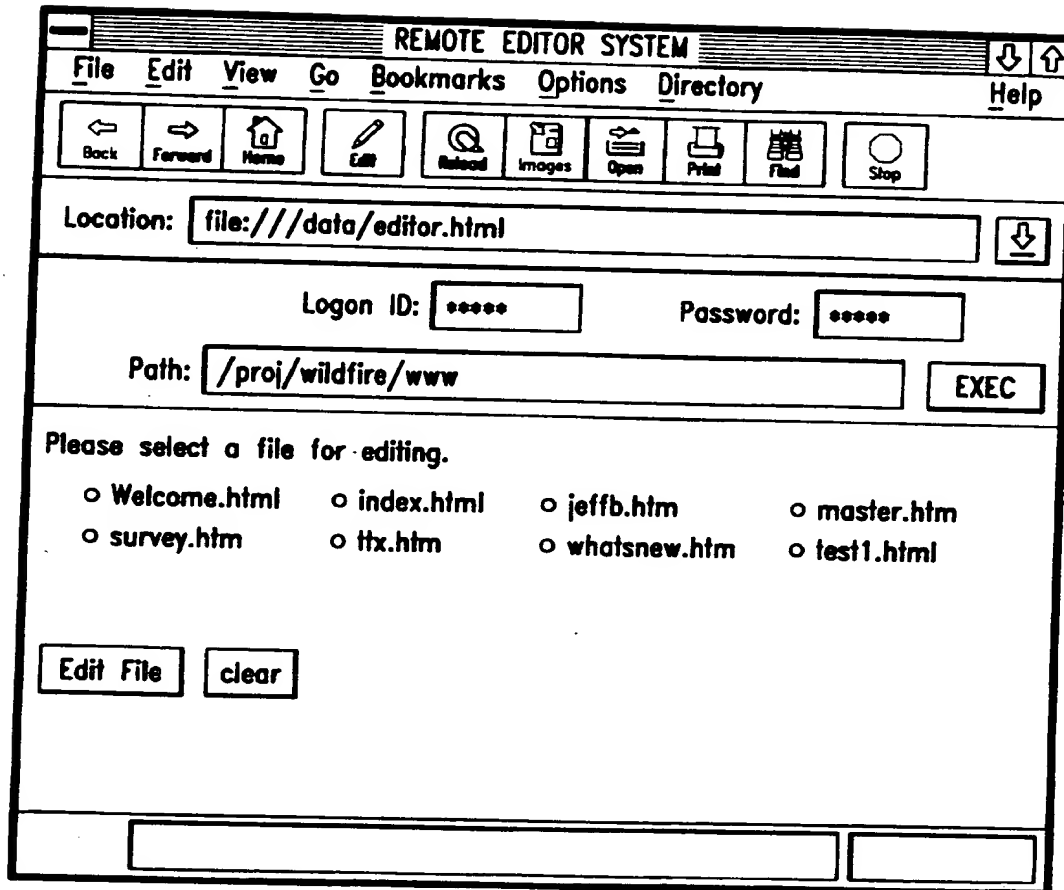


FIG. 4

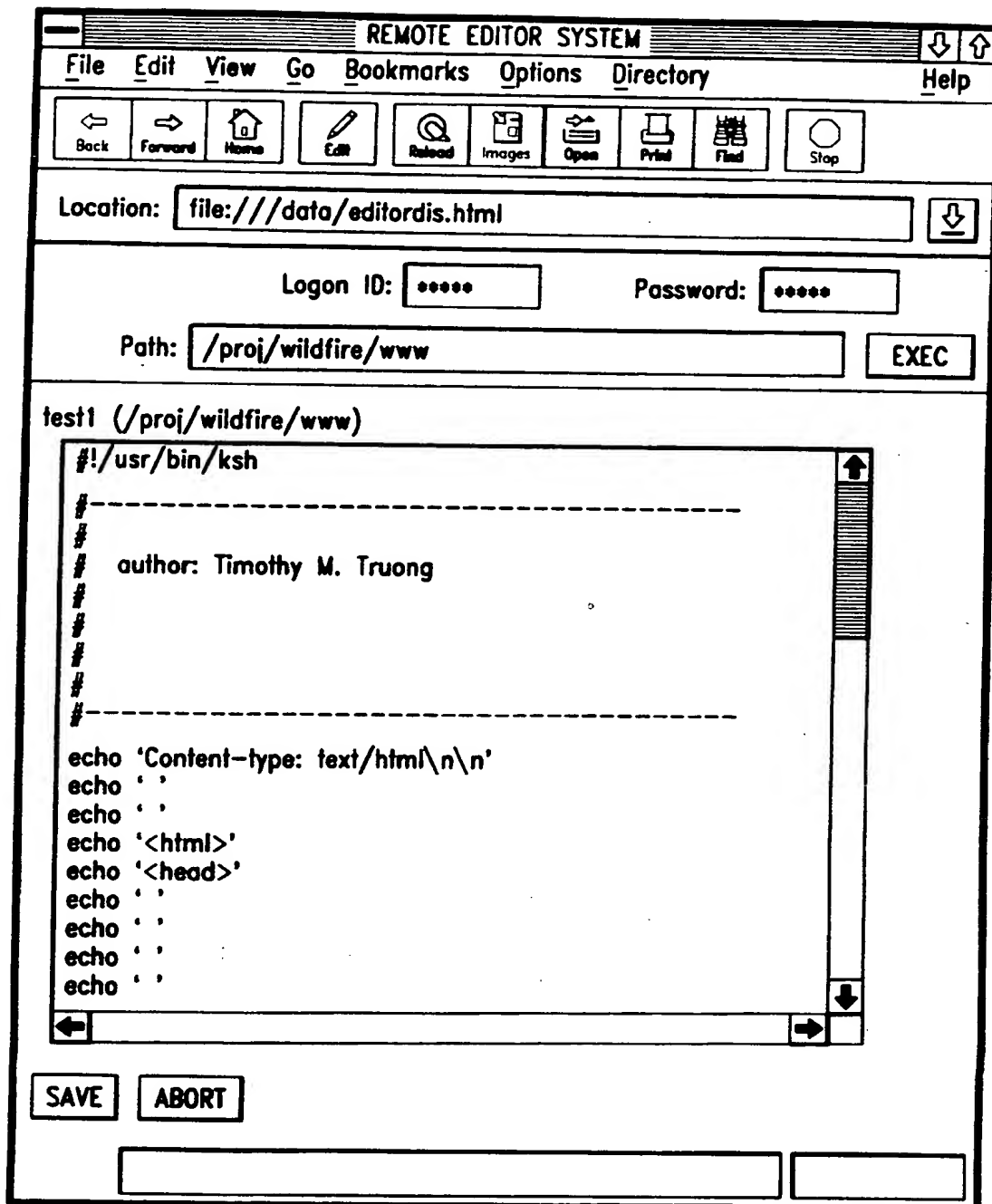


FIG. 5

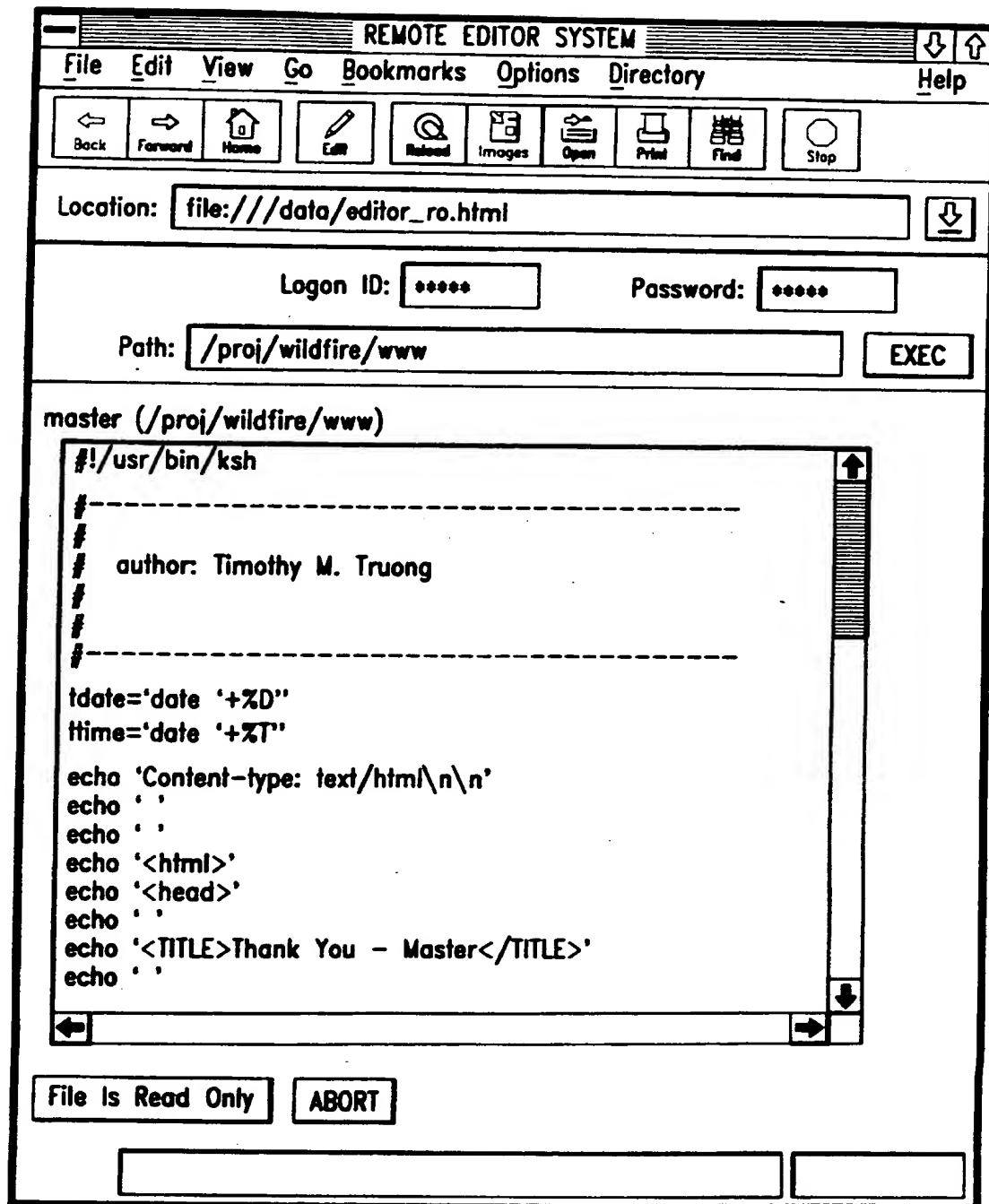


FIG. 6

REMOTE EDITOR SYSTEM

NOTICE

"Copyright 1996 Electronic Data Systems Corporation." A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.

TECHNICAL FIELD OF THE INVENTION

This invention relates generally to the field of computers and more particularly to a remote editor system.

BACKGROUND OF THE INVENTION

Internet growth and use continues to rapidly accelerate as the Internet grows in importance throughout the world. The Internet allows businesses, groups, and individuals to communicate and conduct business electronically. Communication takes place over the Internet in a variety of different ways. For example, businesses and individuals are establishing web pages and home pages that contain a variety of information and are accessible to other Internet users ("users") throughout the world. Users may also communicate with one another through the exchange of files, such as electronic mail ("e-mail") files, and through participation in discussion groups on any number of available topics.

The Internet, which is also referred to as the World Wide Web, is an interconnection of computer servers ("servers") located throughout the world. Internet servers may exchange information or web pages with one another using a protocol known as hypertext transport protocol ("HTTP"). Web pages are stored on the Internet servers and may be communicated to any other server using HTTP. Users may access the Internet from virtually anywhere in the world by interconnecting with one of the Internet servers using a client, such as a personal computer. A user accesses a web page by entering a uniform resource locator ("URL") which identifies a particular web page stored on an Internet server. The term "web page" is used herein to mean any computer file capable of being provided through a computer network to a client, processed, and then displayed. The web page is then communicated from the Internet server, where it is stored, to the server where the client is connected. The web page is then communicated to the client. In this manner, information can be easily disseminated throughout the world. Each web page on the World Wide Web has its own unique URL.

Generally, a user accesses the Internet by executing a "web browser" or "parser" program locally at the client and interconnecting to the server of an Internet service provider. The interconnection may use any of a variety of communication links such as a local telephone communication link or a dedicated communication link. The web browser is a computer program that allows the client to exchange information with the Internet. Any of a variety of web browsers are available, such as NETSCAPE NAVIGATOR, MICROSOFT EXPLORER, NCSA MOSAIC, and others that allow users to conveniently access and navigate the Internet using a graphical user interface. Web browsers receive web pages in a format or language understandable by a web browser, such as Hypertext Markup Language ("HTML"). Web browsers interpret the web pages and generate a corresponding display of the web pages using a graphical user interface.

Some web pages include on-screen forms, including fill-in text boxes, option buttons, radio buttons, and drop-down list boxes that allow a user to interact with a web page. The information provided in these forms may be used as input to a program executing at the server where the web page is stored so that an output web page may be generated in response. NETSCAPE NAVIGATOR, MICROSOFT EXPLORER, and NCSA MOSAIC are known as "forms-capable browsers" because they can interpret HTML code which provides these forms.

Still other web browsers are script-enabled browsers, such as JAVASCRIPT-enabled browsers, and are capable of interpreting HTML web pages that include embedded script within the HTML code. The embedded-script code is interpreted by a script-enabled browser at the client for enhanced processing capability.

Web page designers are able to use the HTML language and scripting languages such as JAVASCRIPT to create web pages or home pages that may be displayed at a client running a web browser. Each web page is assigned a unique address or URL so that users of the Internet may access a desired web page by entering its URL. Many web pages also provide various graphical icons that, if selected, will automatically access another web page. While other web pages include graphical icons that, if selected, will execute a program at a server that generates an output as a web page in HTML format. This output can then be displayed like any other web page. In this manner, users may conveniently navigate the Internet by simply using their mouse and "clicking" on a graphical icon or a link that will automatically take them to a desired web page.

The popularity of the Internet and computers in general has increased the demand for personnel trained in the computer sciences to serve as system administrators. System administrators are responsible for the operation of multiuser computer systems, such as Internet servers. A system administrator of an Internet server is sometimes referred to as a Webmaster. The term system administrator is used herein after to include any user having access or authority to edit a file on a computer or server.

System administrators perform such duties as assigning user accounts and passwords, establishing security access levels, and allocating storage space, as well as being responsible for other tasks such as watching for unauthorized access and preventing virus programs from entering the system. This often involves accessing, viewing, and editing various files of the system. System administrators are also in charge of correcting and recovering from any system failures, such as mass storage failures and memory failures, and returning the system back to normal operation.

Problems arise when system administrators of Internet servers are not physically located at the site of the server and various files of the server need to be accessed, edited, and saved. System administrators have attempted to solve this problem by using communication programs, such as PRO-COMM PLUS by DATASTORM TECHNOLOGIES, INC., and a personal computer with a modem so that the system administrator may directly connect by telephone line to the Internet server and access any file as needed.

This solution has proven troublesome for several reasons. First, the specialized communication software may not be available to the system administrator at all times, such as when the system administrator is on vacation or when the system administrator's personal computer fails. Second, it is often still necessary to set up and initialize communication software at the server and ensure that the communication

17-31
- page
- URL
- HTML

1-50
URL

1-67
browser

software and modem is operational to receive incoming calls. This still requires the physical presence of personnel at the server at a time when personnel may be unavailable. Additionally, remote access using a communication program often involves using a telephone system where expensive long distance telephone fees are incurred. These problems may prevent a system administrator, located away from the server, from accessing and editing files as needed to ensure that the server remains available.

SUMMARY OF THE INVENTION

From the foregoing it may be appreciated that a need has arisen for a remote editor system that allows files stored on an Internet server to be remotely edited. In accordance with the present invention, a remote editor system is provided that allows files stored on an Internet server to be selected and edited or viewed using any client of the Internet using a forms-capable web browser. This eliminates the need to use dedicated communication software to directly access a remote server through a telephone communication link and a dedicated modem. The term "remote location," as used herein, may include any physical distance from a few feet to many thousands of miles and may be a distance across the world, across town, or even across a room.

According to an embodiment of the present invention, there is provided a remote server system using a server coupled to a network that includes a storage medium for storing computer files and a processor responsive to one or more of the computer files of the storage medium. The processor is responsive to communicate an editor input form to a client of the network using a parser program after receiving a request from the client. The processor is then responsive to receive an input text string from the client, that includes a server path input identifying a server path, and stores the server path input as a variable. The processor then generates and communicates an editor selection form to the parser of the client. The editor selection form includes the text of the filenames identifying files included in the server path. After receiving a file selection input identifying one of the files, the processor is responsive to communicate the text of the file to the parser of the client for editing and later saving or storing at the server.

A technical advantage of the present invention includes the ability to remotely access, view, edit, and save a server file using any client having a web browser and connected to the Internet. As the popularity of the Internet increases, Internet connections through a client using a web browser are becoming common place and are frequently found in such places as hotels, libraries, airports, kiosks, and retail stores. This allows a server system administrator to access and edit server files from virtually anywhere in the world without having to have specialized or dedicated communication software available. Another technical advantage of the present invention includes the elimination of costly long distance telephone fees incurred when accessing a server remotely through a long distance telephone connection. Other technical advantages are readily apparent to one skilled in the art from the following figures, description, and claims.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts, in which:

FIG. 1 is a diagram illustrating a network interconnection including a client and an Internet server;

FIG. 2 is a block diagram illustrating the client and a remote Internet server configured as a remote editor system;

FIGS. 3A through 3C depict a flowchart illustrating an exemplary method of using the remote editor system;

FIG. 4 is an exemplary output display of the remote editor system provided to the web browser illustrating the display of a file selection form;

FIG. 5 is an exemplary output display of the remote editor system provided to the web browser illustrating the display of a file for editing; and

FIG. 6 is an exemplary output display of the remote editor system provided to the web browser illustrating the display of a read-only file.

DETAILED DESCRIPTION OF THE INVENTION

Turning first to the nomenclature of the specification, the detailed description which follows is represented largely in terms of processes and symbolic representations of operations by conventional computer components, including a central processing unit ("CPU") or processor associated with a general purpose computer system, memory storage devices for the CPU, and connected pixel-oriented display devices. These operations include the manipulation of data bits by the CPU and the maintenance of these bits within data structures resident in one or more of the memory storage devices. Such data structures impose a physical organization upon the collection of data bits stored within computer memory and represent specific electrical or magnetic elements. These symbolic representations are the means used by those skilled in the art of computer programming and computer construction to most effectively convey teachings and discoveries to others skilled in the art.

For the purposes of this discussion, a process or method is generally considered to be a sequence of computer-executed steps leading to a desired result. These steps generally require manipulations of physical quantities. Usually, although not necessarily, these quantities take the form of electrical, magnetic, or optical signals capable of being stored, transferred, combined, compared or otherwise manipulated. It is conventional for those skilled in the art to refer to these signals as bits, values, elements, symbols, characters, text, terms, numbers, records, files, or the like. It should be kept in mind, however, that these and some other terms should be associated with appropriate physical quantities for computer operations, and that these terms are merely conventional labels applied to physical quantities that exist within and during operation of the computer.

It should also be understood that manipulations within the computer are often referred to in terms such as adding, comparing, moving, etc., which are often associated with manual operations performed by a human operator. It must be understood that no involvement of a human operator is necessary or even desirable in the present invention. The operations described herein are machine operations performed in conjunction with a human operator or user that interacts with the computer or computers.

In addition, it should be understood that the programs, processes, methods, etc. described herein are but an example of one implementation of the present invention and are not related or limited to any particular computer, apparatus or computer language. Rather, various types of general purpose computing machines or devices may be used with programs

constructed in accordance with the teachings described herein. Similarly, it may prove advantageous to construct a specialized apparatus to perform the method steps described herein by way of dedicated computer systems with hard-wired logic or programs stored in non-volatile memory, such as read-only memory.

Referring now in more detail to the drawings, FIG. 1 is a diagram illustrating a network interconnection 10 including a client 12 and an Internet server 14. Network interconnection 10 includes the interface between Internet server 14 and a plurality of clients through a plurality of direct communication lines 18. For example, network interconnection 10 includes an interconnection between client 12 and Internet server 14 through a direct communication line 16. Network interconnection 10 also includes the interface between Internet server 14 and other Internet servers through a plurality of communication lines 20. In this manner, Internet server 14 allows the plurality of clients directly interconnected with server 14, such as client 12, to communicate with other Internet users by providing an interconnection to other Internet servers through communication lines 20. The other Internet users are also directly interconnected with a server such that a communication path may be established between one client and another client through the various servers of the Internet. Internet server 14 may be used by a local Internet service provider to allow users to access the Internet.

Internet server 14 is a computer such as a personal computer, file server, workstation, minicomputer, mainframe, or any other computer capable of communicating and interconnecting with other computers. Internet server 14 will preferably include a processor, a printer, an input device such as a mouse and/or a keyboard, a monitor, a floppy disk drive, memory, a modem, and a mass storage device such as a hard disk drive. Communication lines 20 and direct communication lines 18 may be any type of communication link capable of supporting data transfer. For example, these communication lines may include any combination of an Integrated Services Digital Network ("ISDN") communication line, a hard wired line, or a telephone link.

Client 12 may be similar to Internet server 14 and may be implemented using virtually any type of computer. Client 12 will preferably be a personal computer having a processor, a printer, an input device such as a mouse and/or a keyboard, a monitor, a floppy disk drive, memory, a modem, and a mass storage device such as a hard disk drive. Client 12 and Internet server 14 will be operating under the control of an operating system such as MS-DOS, Macintosh OS, WINDOWS NT, WINDOWS 95, OS/2, UNIX, XENIX, and the like. Client 12 and Internet server 14 may execute any number of available application programs such as a web browser.

In operation, the various clients of network interconnection 10, such as client 12, may communicate through server 14 with any other client connected to the Internet. For example, client 12, generally using a web browser application program, may couple to Internet server 14 and provide the address or URL of an Internet web page. The Internet web page will generally be stored on another Internet server located anywhere in the world. Internet server 14, through the plurality of communication lines 20, communicates with other Internet servers using HTTP and provides the URL to other Internet servers. Eventually, the receiving Internet server is found and in response, transmits the Internet web page back to Internet server 14 for viewing by client 12 using a web browser.

Although network interconnection 10 has been illustrated and described in FIG. 1 as being a node or interconnection on the Internet, network interconnection 10 may be any interconnection found on any computer network such as a local area network ("LAN"), a wide area network ("WAN"), an intranet, such as a corporate intranet, or any other communications and data exchange system created by connecting two or more computers. The present invention will be illustrated and described with an implementation using the Internet, however, it should be understood that the present invention is not limited to only implementations using the Internet.

FIG. 2 is a block diagram illustrating client 12 and a remote Internet server 15 configured as a remote editor system 26. Client 12 includes a processor 34, input/output ("I/O") devices 36, and a client memory 48. Processor 34, under the control of an operating system 30, controls the operation of client 12 and is used to retrieve, process, store, and display data. Operating system 30 and a web browser 32 are stored in client memory 48. A logon script 41 may also be stored in client memory 48 and is provided as an embedded script from an Internet web page. Logon script 41 is discussed more fully below. Client memory 48 may be a random access memory ("RAM").

Processor 34 is typically implemented as a microprocessor, such as those manufactured by INTEL or MOTOROLA. Processor 34 may include an arithmetic logic unit to assist in performing mathematical operations. Processor 34 communicates control, address, and data signals with operating system 30 and with the remaining components of client 12 through a system bus. Processor 34 interprets and executes instructions that have been fetched or retrieved from client memory 48 and may be implemented as a single integrated circuit or as a combination of integrated circuits.

I/O devices 36 may include any peripheral that allows data to be exchanged with client 12 and may include such devices as a keyboard, a monitor, a printer, a modem, a pointing device, such as a mouse, a floppy disk drive, a mass storage device, such as a hard disk drive, and the like. The mass storage device is used to store computer files including application programs and data files. For example, the mass storage device may be used to store web browser 32 and operating system 30.

Operating system 30 includes a set of computer programs that control the internal functions of client 12, thereby allowing client 12 to run application programs. Operating system 30 is typically stored in and provided from a mass storage device, such as a hard disk drive, a floppy disk drive, a CD ROM drive or a ROM chip. During start-up or initialization of client 12, operating system 30 is loaded into client memory 48. Application programs, such as web browser 32, may also be loaded into client memory 48 along with operating system 30.

Web browser 32 is preferably a graphical web browser or parser that allows a user to view images, fonts, and document layouts provided in a web page by converting large units of data into smaller, more easily interpreted, units of data. Web browser 32 reads the tagged text of a web page provided in HTML format. HTML uses tags to identify the parts of a web page, such as headings, bulleted lists, body text, on-screen forms, including fill-in text boxes, option buttons, radio buttons, and drop-down list boxes, images to be displayed, hypertext links, colors, font, and various other formatting tags. Web browser 32 formats the various parts of the document for on-screen display as directed by the HTML tags.

1-27
client
server
line

1-27
client
server
line

41-53

4-61
browser
address
or
URL

55-67
-HTML
-browser

Web browser 32 is also a script-enabled browser which enables it to interpret HTML formatted web pages that include embedded script within the HTML code. The embedded script is provided to web browser 32 at client 12 for enhanced processing at client 12. The embedded script may be provided in JAVASCRIPT format or any other scripting language format that is provided to a web browser at a client for enhanced processing.

Web browser 32 operates in conjunction with operating system 30 so that client 12 may properly interface with the Internet. Web browser 32 provides a graphical user interface and is accessed by operating system 30 at the request of a user. Once client 12 is coupled to Internet server 14 through direct communication line 16, as shown in FIG. 1, web pages may be accessed by entering the URL of the desired web page into web browser 32. It should be noted that web browser 32 is not a terminal emulation program or communication program that is used at client 12 to emulate a server terminal operating directly from remote Internet server 15.

Once a desired web page is retrieved, web browser 32 may receive formatting information and embedded script from a file defining the web page. The file defining the web page is generally located at a remote Internet server, such as remote Internet server 15 as shown in FIG. 2. Typically, web browser 32 receives the information in HTML format and the embedded script in JAVASCRIPT format so that the web page may be interpreted and processed using processor 34 and web browser 32 of client 12 and then graphically displayed at client 12. Often, a web page will contain user selectable icons that are preprogrammed with the URL of a related web page so that a user may conveniently navigate the Internet by selecting these icons.

Remote Internet server 15 includes a server memory 46, a processor 42, and a mass storage device 44. Although not shown in FIG. 2, remote Internet server 15 also preferably includes various I/O devices such as those mentioned above in the description of I/O devices 36 of client 12. Operating system 38 and remote editor program 40 are stored in mass storage device 44 and are shown loaded into server memory 46.

Processor 42, in conjunction with operating system 38, controls the operation of remote Internet server 15. Processor 42 fetches and executes various instructions stored in server memory 46. Operating system 38 operates similarly to operating system 30 of client 12 and includes a set of computer programs that control the internal functions of remote Internet server 15. Operating system 38 controls the allocation and usage of hardware resources such as server memory 46, processor 42, mass storage device 44, and optional I/O devices (not specifically shown). Operating system 38 may be any of a variety of available operating systems depending partially on the hardware of server 14. For example, operating system 38 may be implemented as MS-DOS, the Macintosh OS, OS/2, WINDOWS NT, WINDOWS 95, UNIX, XENIX or any of a variety of other operating systems.

Remote editor program 40 is an application program shown loaded into server memory 46. Remote editor program 40 is stored in mass storage device 44 and is then loaded into server memory 46 when selected by a user. This may occur when a user of client 12, while accessing the Internet using web browser 32, requests a particular web page that will automatically load remote editor program 40 into server memory 46. In response, HTML code and any embedded script may be provided to web browser 32. For example, logon script 41, as shown in client memory 48 with

web browser 32, may be provided to client 12 where it is processed using web browser 32 and processor 34.

Remote editor program 40, discussed more fully below in connection with FIGS. 3A through 3C, 4, 5, and 6, communicates an editor input form, including HTML code and embedded script code, to web browser 32 of client 12. In response, client 12 preferably prompts the user for an input including a logon ID, a password, and a remote server path using a web page, such as that shown in FIGS. 3A through 3C. Remote editor program 40 may be implemented using a common gateway interface ("CGI") program, called a script, that receives the input from web browser 32 of client 12, processes the input and executes other programs of remote Internet server 15 as necessary, and provides any results to web browser 32 in HTML format. Appendix A includes some sample code of one such implementation of the editor input form of remote editor program 40.

The logon ID, password, and remote server path inputs identify a particular user and whether the user has access rights to remote Internet server 15. Logon script 41, provided by remote editor program 40 through the editor input form, is stored in client memory 48 and is interpreted by web browser 32 to determine whether text has been entered into the logon ID input and the password input. Assuming that text has been entered, the inputs are communicated back to remote editor program 40 of remote Internet server 15 as an input text string. Remote editor program 40 receives the input text string from the client and parses the input text string to identify a server path input. This server path input identifies a particular path at remote Internet server 15. Assuming that the user provided a valid logon ID and password, as provided through the input text string, remote editor program 40 stores the server path input as a variable and generates a file selection form. The file selection form includes the text of the file names identifying files included in the server path defined by the server path input. Remote Internet server 15, under the control of remote editor program 40, communicates the file selection form to the client in HTML format.

Web browser 32 of client 12 receives the file selection form and provides a display such as that shown in FIG. 4. The user then selects the desired file for editing by selecting the radio button corresponding to the desired file. Once selected, the user selects the execute button and web browser 32 of client 12 provides a file selection input to remote Internet server 15. The file selection input identifies the file name of the file selected by the user. In response, remote Internet server 15, under the control of remote editor program 40, communicates the text of the file to web browser 32 of client 12 for editing. The file may then be edited and sent back to remote Internet server 15 where the edited text is stored under the selected file name. In this manner, a system administrator located away from Internet server 15 may access, view, and edit any file stored at Internet server 15.

The present invention may be implemented using any of a variety of computer languages and using any of a variety of computer hardware operating under any of a variety of operating systems. For example, remote Internet server 15 may be a work-station operating under the UNIX operating system. The UNIX operating system supports shell scripts, such as Korn shells, that may be used to provide various programming features and to provide HTML information to a web browser. Korn shells function as CGI scripts and may also be used to call other routines such as C programs. The Korn shell serves as a CGI program and provides HTML formatted information to web browser 32 so that the output of remote editor program 40 may be displayed as a web page.

FIGS. 3A through 3C depict a flow chart illustrating an exemplary method of using remote editor system 26. The method begins at step 100 and proceeds to step 102 where a web browser application program is executed at a client and the client in turn connects to the server of an Internet service provider. At step 104 a user enters the URL of a network editor web page stored on a remote Internet server. Next, at step 106 the remote Internet server provides the network editor web page to the web browser at the client. The network editor web page may also be referred to as an editor input form and preferably includes a first part, generally provided in HTML format to the web browser or the client, and is used to generate an input screen at the client. The network editor web page preferably also includes a second part, generally provided in JAVASCRIPT format, that is provided to the web browser of the client so that the client may perform local processing such as logon script 41 of FIG. 2. The network editor web page includes HTML code and JAVASCRIPT code.

The method proceeds next to step 108 where the web browser displays the network editor web page at the client. The network editor web page provides input fields for a logon ID input, a password input, and a remote server path input such as that shown in FIGS. 4, 5, and 6. Then at step 110 a user provides a logon ID, a password, and a remote server path as an input at the web browser of the client.

At step 112 the web browser, using the embedded logon script provided in step 106, performs processing to ensure that the logon ID and password fields are not empty. The logon script simply checks to ensure that something has been entered in both of these inputs. At decision step 114 it determines whether the logon ID input or the password input are empty. If so, at step 116, a message is displayed at the web browser of the client indicating that a valid logon ID and password must be entered to access the system. The method then returns to step 108 where the user has the opportunity to enter a valid logon ID and password. Otherwise, the method at decision step 114 proceeds to step 118.

Step 118 involves sending the logon ID, password, and remote server path inputs to the remote Internet server in the form of an input text string. The server receives the input text string at step 120 and determines whether too many characters have been provided by the user at decision step 122. If too many characters have been provided in the input text string, the method proceeds to step 124. Step 124 involves displaying a message at the client indicating to the user that the text string is too long and that shorter inputs, in particular the remote server path, must be provided. The method then proceeds back to step 108. Otherwise, the method at decision step 122 proceeds to step 126.

Step 126 involves parsing the input text string into a logon ID string, a password string, and a remote server path string. The method then proceeds to step 128 where each string is stored as a variable. Next, the method proceeds to step 130 where the logon ID and password variables are compared to valid logon IDs and valid passwords to determine if both of these inputs are valid. This may be done by comparing the logon ID variable and the password variable to entries in a file or to other variables storing valid logon IDs and passwords. The method proceeds next to decision step 132. Decision step 132 proceeds to step 134 if the logon ID and the password are determined not to be valid. Step 134 displays a message at the client indicating that the user is not authorized to use this system. The method then returns to step 108 where the user may enter a valid logon ID and password. After a predetermined number of attempts, the

method may proceed to step 168 where the method ends. On the other hand, if the logon ID and password are determined to be valid in decision step 132, the method proceeds to step 136.

Step 136 and step 138 involve generating file name text for each server file stored in the directory identified by the server path input variable and providing this text to the web browser of the client along with code to generate a radio button input next to the text of each file name at the web browser. The text as provided to web browser in these steps may be referred to as a file selection form. At step 140 the web browser of the client displays the text of each filename and the corresponding radio button as illustrated in the exemplary output display of FIG. 4.

The method proceeds next to step 142 where a user selects the desired file for viewing/editing by selecting the radio button corresponding to the desired filename. At this point the method proceeds to step 144 where the filename of the selected file is sent to the server. The filename may be sent as a text string and referred to as a file selection input which identifies the selected file. The server parses the input string of the file selection input in step 146. Next, at step 148 a search is performed on the selected file at the server to see whether the file contains certain code or text, such as HTML tags, that would interfere with the web browser at the client when interpreting this file. Step 148 also involves looking at the attributes of the selected file to determine if the file is a read-only file. The method proceeds next to decision step 150 where if the file cannot be displayed by web browser, the method proceeds to step 152 where a message is displayed at the client indicating that the selected file is not capable of being viewed by the web browser. The method then would return to step 108. If decision step 150 determines that the file may be displayed by web browser, the method proceeds to decision step 154.

The method proceeds next to step 156 if decision step 154 determines that the file is a read-only file. Step 156 involves displaying a message to the web browser at the client indicating that the selected file is a read-only file. Then at step 158, the text of the read-only file is provided to the web browser for viewing as illustrated in FIG. 6. The method then proceeds to step 108. However, if decision step 154 determines that the selected file is not a read-only file, the method proceeds instead to step 160.

Step 160 involves providing the text of the selected file to the web browser for editing as shown in FIG. 5. The method then proceeds to step 162 where the text may be edited at the web browser using editing features such as delete, select, search, copy, paste, and the like. The text may be scrolled up/down and right/left using the provided arrows. After the text has been edited in step 162, the method proceeds to decision step 164. Decision step 164 determines whether or not the abort button has been selected, as shown in FIG. 5. If the abort button is selected, the method returns to step 108; otherwise, the method proceeds to step 166 where the edited text is sent from the client to the server and saved under the selected filename. The method ends at step 168.

FIG. 4 is an exemplary output display of remote editor system 26 provided to web browser 32 illustrating the display of a file selection form. At the point when the file selection form is displayed, the user has already entered the URL of the remote editor web page in the Location field and has provided a valid logon ID, password, and remote server path. After entering the logon ID, password, and path, the user has selected the "EXEC" button and remote editor system 26 as responded with the file selection form listing

15-35
filename

-26

the eight files and corresponding radio buttons as shown in FIG. 4. At this point, the user may select a file for editing or viewing by selecting the radio button corresponding to the desired file. After selecting the radio button, the edit file button is selected and remote editor system 26 responds with the display as shown in FIG. 5 or FIG. 6 depending on whether the selected file is displayable by web browser 32 and whether the file is a read-only file.

FIG. 5 is an exemplary output display of remote editor system 26 provided to web browser 32 illustrating the display of a selected file for editing. In this case, remote editor system 26 has provided the "test1" file for editing after determining that this file may be viewed by web browser 32 and the file is not a read-only file. A user may now edit the text of the file as desired. Assuming that remote editor system 32 is running under the WINDOWS operating system, any of the basic WINDOWS editing commands such as a copy, insert, delete, and paste may be used when editing the file.

FIG. 6 is an exemplary output display of remote editor system 26 provided to web browser 32 illustrating the display of a read-only file. If remote editor system 26 determines that the selected file is viewable by a web browser but is a read-only file, remote editor system 26 provides the file, such as the "master" file as shown in FIG. 6, for viewing by the user. The user may use the up/down and left/right arrows to navigate the document. When finished, the user may select the abort button or may enter a different path in the path field and select the "EXEC" button to retrieve a new file selection form for a different path of remote Internet 15. Remote editor system 26 displays a message that the file is read-only file with the text shown towards the bottom of FIG. 6.

Thus, it is apparent that there has been provided, in accordance with the present invention, a remote editor system that allows a file to be edited from a remote location that satisfies the advantages set forth above. Although the preferred embodiment has been described in detail, it should be understood that various changes, substitutions, and alterations can be made herein. For example, the present invention may be implemented on any computer, computer network, or interconnection of computers. Also, the direct interconnections illustrated herein could be altered by one skilled in the art such that two computers or devices are merely coupled to one another through an intermediate computer or device without being directly connected while still achieving the desired results demonstrated by the present invention. Although the present invention has been primarily described and shown as implemented on the Internet, it should be understood that the present invention is not limited to the Internet and can include a variety of other computer networks such as an intranet. Other examples of changes, substitutions, and alterations are readily ascertainable by one skilled in the art and could be made without departing from the spirit and scope of the present invention.

APPENDIX A

```
<HTML>
<HEAD>
<TITLE>Remote Editor System</TITLE>
<BASE HREF = "http://www.sets">
<META NAME = "AUTHOR" CONTENT = "tim">
</HEAD>
<frameset rows = "10%, *">
```

APPENDIX A-continued

```
<frame name = "headframe" src =
"file:///c:/patent/texthead.htm" scrolling="no">
<frame name = "bodyframe" src =
"file:///c:/patent/textbody.htm">
</frameset>
</HTML>
<HTML>
10 <HEAD>
<TITLE>Remote Editor System</title>
<BASE HREF = "http://www.sets">
<META NAME = "AUTHOR" CONTENT = "tim truong">
</HEAD>
<BODY background = ". /papers/chalk.xxx">
15 </BODY>
</HTML>
<HTML>
<HEAD>
<TITLE>Remote Editor System</title>
<BASE HREF = "http://www.sets">
20 <META NAME = "AUTHOR" CONTENT = "tim truong">
<script language = "javascript">
<!-- to hide script contents from old browsers
function MakeArray (n)
{
25   this.length = n
   return this
}
pathNames as new MakeArray(2)
pathNames [0] as "/proj/wildfire/www/httpd/cgi-bin/PRC-cgi-bin"
pathNames [1] = "/proj/wildfire/www/_doc/sets/PRC"
function fillDir (form)
30 {
   for (var i = 0; <2; i++)
   {
      if (form.dirname[i].checked)
      {
         form.dirpath.value = pathNames [i];
         form.dirpath.focus();
      }
   }
}
function runpl (form,button)
40 {
   if form.idtext.value.length == 0 )
   {
      alert ("Need to have ID to process request.");
      form.target = "headframe";
      form.action = "file:///c:/patent/texthead.htm";
   }
}
// end hiding contents from old browsers
45 </script>
</HEAD>
<BODY background = ". /papers/chalk.xxx">
<form method = "post" action = "/PRC-cgi-bin/text" target =
"bodyframe">
<center>
50 <table border = 0>
<tr>
<td align = center valign = middle>
   Logon ID: <input type = "text" name = "idtext"
   size = 9>
   Password: <input type = "password" name =
   "passwdtext" size = 10>
55 </td>
<td align = center>
   <input type = "radio" name = "dirname" value =
   0 onclick = "fillDir(this.form)"> CGI
   <input type = "radio" name = "dirname" value =
   1 onclick = fillDir(this.form)"> PRC
   <input type = "text" name = "dirpath" size =
   40>
   </td>
<td>
   <input type = "submit" name = "exec" value =
   "EXEC" onclick =
   "runpl(this.form,this.button)">
60 </td>
65 </tr>
```

APPENDIX A-continued

```

</tr>
</table>
</form>
<br>
<br>
</BODY>
</HTML>
# -----
# author: Timothy M. Truong
#
# -----
echo "Content-type: text/html;"
echo ""
echo ""
echo "<html>"
echo "<head>"
echo ""
echo "<TITLE>Remote Editor System</title>"
echo "<base href='http://www.seta/'>"
echo "<meta name='author' content='tim truong'>"
#echo "<script language='livescript'>"
#echo "<!-- to hide from old browsers -->"
#echo "</script>"
echo "</head>"
echo "<body background='./paper/chalk.png'>"
# ----- determine method -----
if [ $REQUEST_METHOD = "GET" ]; then
  BUFF=$QUERY_STRING
else
  LENGTH=$CONTENT_LENGTH
  if [ $LENGTH -gt 32000 ]; then
    echo "<hr>"
    echo "<h3> Sorry, you had sent a too long text</h3>"
    echo "<hr>"
    echo "</body>"
    echo "</html>"
    exit
  fi
  read BUFF
fi
# ----- parse input -----
echo $BUFF |
  /proj/wildfire/paser1.pl |
while read line
do
  field_name= echo $line | cut -f1 -d=
  field_value= echo $line | cut -f2 -d=
  case $field_name in
    idtext)
      id="$field_value"
      ;;
    passwdtext)
      Passw="$field_value"
      ;;
    dirpath)
      Dir="$field_value"
      ;;
    esac
done
# ----- check security -----
if [ "$id" != "tmt" ] || [ "$Passw" != "tmt" ]; then
  echo "<hr>"
  echo "<h3> Sorry, you are not allowed to use this
function.</h3>"
  echo "<hr>"
  echo "</body>"
  echo "</html>"
  exit
fi
# ----- send stuff to browser -----
echo "<form method='post' action='/PRC-cgi-bin/text1'
target='bodyframe'>"
echo "<h2>Please select a file for editing.</h2>"
echo "<p>"
echo "<table border=0 cellpadding=10>"
echo "<tr>"

```

APPENDIX A-continued

```

Index=1
cd "$Dir"
5 for i in `ls`; do
  if [ $Index = 10 ]; then
    echo "</tr>"
    echo "<tr>"
    Index=1
  fi
  10 PType= file $i | awk '{print $2}'
  case $PType in
    directory)
      ;;
    *)
      15 echo "<td>"
      echo "<input type='radio' name='filename' value='`$i`'>`$i`"
      echo "</td>"
      ;;
    esac
    Index = `expr $Index + 1`
  done
  echo "</tr>"
  echo "</table>"
  echo "<hr>"
  echo "<input type='submit' value='Edit File'>"
  echo "<input type='reset' value='clear'>"
  25 echo "<input type='hidden' value='$Dir' name='dirpath'>"
  # ----- closing tags -----
  echo "</form>"
  echo "</body>"
  echo "</html>"
  # -----
  30 #
  # author: Timothy M. Truong
  #
  # -----
  echo "Content-type: text/html;"
  echo ""
  35 echo ""
  echo "<html>"
  echo "<head>"
  echo ""
  echo "<TITLE>Remote Editor System</title>"
  echo "<base href='http://www.seta/'>"
  echo "<meta name='author' content='tim truong'>"
  40 #echo "<script language='livescript'>"
  #echo "<!-- to hide from old browsers -->"
  #echo "</script>"
  echo "</head>"
  echo "<body background='./paper/chalk.png'>"
  45 echo "<form method='post' action='/PRC-cgi-bin/text2'
  target='bodyframe'>"
  # ----- determine method -----
  if [ $REQUEST_METHOD = "GET" ]; then
    BUFF=$QUERY_STRING
  else
    50 LENGTH=$CONTENT_LENGTH
    if [ $LENGTH -gt 32000 ]; then
      echo "<h3> Sorry, you had sent a too long text</h3>"
      echo "<hr>"
      echo "</body>"
      echo "</html>"
      exit
    fi
    55 read BUFF
  fi
  echo $BUFF > /tmp/kill
  # ----- parse input -----
  echo $BUFF |
  60 /proj/wildfire/paser1.pl |
  while read line
  do
    field_name= echo $line | cut -f1 -d=
    field_value= echo $line | cut -f2 -d=
    case $field_name in
      65 dirpath)
        Dir="$field_value"
      ;;
    esac
  done

```

APPENDIX A-continued

```

    ;;
    filename)
        FileName=$field_value
    ;;
esac
done
# ----- process -----
echo "<h2>$FileName ($DirPath)</h2>"
cd $DirPath
Result=$(grep -ci '<textarea>' "$FileName")
if [ $Result != 0 ]; then
    echo "<hr>"
    echo "<h3>Sorry, the file you selected contains data that
conflict"
    echo "with this program thus makes it unviewable and
unreadable."
    echo "</h3>"
    echo "<hr>"
    echo "</body>"
    echo "</html>"
    exit
else
    echo "<pre>"
    echo "<textarea name='filedata' rows=20 cols=80>"
    cat "$FileName"
    echo "</textarea>"
    echo "</pre>"
    echo "<hr>"
    if [ -w "$FileName" ]; then
        echo "<input type='submit' value='SAVE'>"
    else
        echo "<input type='button' value='File Is Read Only'>"
    fi
    echo "<input type='reset' value='ABORT'>"
    echo "<input type='hidden' value='$DirPath' name='dirpath'>"
    echo "<input type='hidden' value='$FileName' name='filename'>"
    echo "</form>"
    # ----- closing tags -----
    echo "</body>"
    echo "</html>"
    #
    # author: Timothy M. Truong
    #
    # -----
    echo "Content-type: text/html; charset=utf-8"
    echo ""
    echo "<html>"
    echo "<head>"
    echo ""
    echo "<TITLE>Remote Editor System</title>"
    echo "<base href='http://www.ssta/'>"
    echo "<meta name='author' content='tim truong'>"
    echo "<script language='javascript'>"
    echo "<!-- to hide from old browsers -->"
    echo "// end hiding -->"
    echo "</script>"
    echo "</head>"
    echo "<body background='./paper/chalk.png'>"
    echo "<form method='post' action='/PRC/cgi-bin/tex2'
target='bodyframe'>"
    # ----- determine method -----
    if [ $REQUEST_METHOD = "GET" ]; then
        BUFF=$QUERY_STRING
    else
        LENGTH=$CONTENT_LENGTH
        if [ $LENGTH -gt 32000 ]; then
            echo "<h3>Sorry, you had sent a too long text</h3>"
            echo "<hr>"
            echo "</body>"
            echo "</html>"
            exit
        fi
        read BUFF
    fi
    echo $BUFF > /tmp/kill
    # ----- parse input -----

```

APPENDIX A-continued

```

    echo $BUFF |
    /proj/wildfire/parser2.pl |
5   while read line
    do
        field_name=$(echo $line | cut -f1 -d=)
        field_value=$(echo $line | cut -f2 -d=)
        case $field_name in
            dirpath)
                Dir="$field_value"
                ;;
            filename)
                FileName="$field_value"
                ;;
            filedate)
                FileData="$field_value"
15         ;;
        esac
    done
    # ----- process -----
    cd $DirPath
    cat /tmp/tex2.$FileData > "$DirPath/$FileName"
    echo "<hr>"
    echo "<p>File <b> '$DirPath/$FileName' </b> updated
successful</h1>"
    echo "<hr>"
    # ----- closing tags -----
    echo "</body>"
    echo "</html>"
25   yes | rm /tmp/tex2.$FileData
    #
    # author: Timothy M. Truong
    #
    # -----
    $formData = <STDIN>;
    chop ($formData);
    chop ($formData);
    # ----- parse input -----
    foreach (split(/&/, $formData ))
35   {
        ($fieldName, $fieldValue) = split (/=/, $__);
        $fieldName = s/\+//g;
        $fieldName = s/%([0-9A-F] {2})/pack (C, hex($1))/eg;
        $fieldValue = s/\+//g;
        $fieldValue = s/%([0-9A-F] {2})/pack (C, hex($1))/eg;
        $myData{$fieldName} = $fieldValue;
40   }
    # ----- assign values -----
    foreach $term (keys(%myData))
    {
        print ($term, "=", $myData{$term}, "\n");
45   }
    #
    # author: Timothy M. Truong
    #
    # -----
    unless (open (OUTfile, ">/tmp/tex2.$$"))
    {
        die (" ");
    }
    $formData = <STDIN>;
    chop ($formData);
    # ----- parse input -----
55   foreach (split(/&/, $formData ))
    {
        ($fieldName, $fieldValue) = split (/=/, $__);
        $fieldName = s/\+//g;
        $fieldName = s/%([0-9A-F] {2})/pack (C, hex($1))/eg;
        $fieldValue = s/\+//g;
60   if ($fieldName ne "filedata")
        {
            $fieldValue = s/%([0-9A-F] {2})/pack (C, hex($1))/eg;
            chop($fieldValue);
        }
        else
65   {
            $fieldValue = s/%0D%0A//g;

```

APPENDIX A-continued

```

$fieldValue = "u/%(0-9|A-F){2}/pack(C,hex($1))/eg;
print OUTfile ($fieldValue);
$fieldValue = "$$";
}
$myData{$fieldName} = $fieldValue;
#----- assign values -----
foreach $term (keys(%myData))
{print ($term,"-", $myData{$term},"n");}

```

What is claimed is:

1. A remote system administration method, comprising:
 - receiving an editor input form at a forms-enabled and script-enabled web browser through a network from a server in response to a request from a client, the web browser resident on the client;
 - sending a server path input to the server from the web browser;
 - receiving a file selection form from the server, the file selection form including filenames identifying files included in a server path defined by the server path input;
 - sending a file selection from the web browser to the server, the file selection identifying one of the files;
 - receiving a copy of the one of the files from the server;
 - editing the copy of the one of the files using the web browser without the use of a plug-in to the web browser to produce an updated file; and
 - sending the updated file to the server for storage.
2. The method of claim 1, further comprising saving the updated file at the server.
3. The method of claim 1, further comprising performing system failure recovery operations in response to the updated file.
4. The method of claim 1, further comprising using the updated file to perform one of the group consisting of assigning a user account, identifying a security access level, allocating storage space, and monitoring unauthorized access.
5. The method of claim 1, further comprising receiving an indication at the client that selected text is present in the one of the files.
6. The method of claim 1, further comprising receiving a message at the client indicating that the one of the files is a read-only file and receiving the copy of the one of the files at the web browser for viewing.
7. A remote system administration method, comprising the steps of:
 - communicating an editor input form from a server through a network to a client in response to receiving a request from the client, the client using a forms-enabled and script-enabled web browser;
 - receiving a server path input at the server from the web browser;
 - communicating a file selection form from the server to the web browser, the file selection form including filenames identifying files included in a server path defined by the server path input;
 - receiving a file selection from the web browser at the server, the file selection identifying one of the files; and
 - communicating a copy of one of the files from the server to the web browser for editing;

receiving by the server an updated file for storage, the updated file produced by editing the copy of the one of the files using the web browser without the use of a plug-in to the web browser.

8. The method of claim 7, further comprising the step of saving the updated file at the server.
9. The method of claim 7, further comprising using the updated file to perform one of the group consisting of performing system failure recovery operations, assigning a user account, identifying a security access level, allocating storage space, and monitoring unauthorized access.
10. The method of claim 7, further comprising searching the one of the files to determine if selected text is present.
11. The method of claim 7, further comprising communicating a text message to the client indicating that the one of the files is not viewable by the web browser.
12. The method of claim 7, further comprising analyzing the one of the files to determine if the one of the files is a read-only file, communicating a message to the client indicating that the one of the files is a read-only file and communicating the copy of the one of the files to the web browser for viewing.
13. A remote system administration editor system, comprising:
 - a storage medium for storing at least one computer file; and
 - a server responsive to one or more of the computer files of the storage medium to
 - communicate an editor input form to a client using a network in response to receiving a request from the client, the client using a forms-enabled and script-enabled web browser;
 - receive a server path input in response to the editor input form from the web browser;
 - communicate a file selection form to the web browser that includes the text of the filenames identifying files included in a server path defined by the server path input;
 - receive a file selection in response to the file selection form, the file selection identifying one of the files;
 - communicate a copy of the one of the files associated with the file selection to the web browser for editing; and
 - receive an updated file for storage, the updated file produced by editing the copy of the one of the files that has been edited with the web browser without the use of a plug-in to the web browser.
14. The system of claim 13, wherein the server is responsive to save the updated file.
15. The system of claim 13, wherein the server is responsive to perform one of the group consisting of performing system failure recovery operations, assigning a user account, identifying a security access level, allocating storage space, and monitoring unauthorized access in response to the updated file.
16. The system of claim 13, wherein the editor input form and the file selection form are provided to the web browser in hypertext markup language format.
17. The system of claim 13, wherein the server is responsive to search the one of the files to determine if selected text is present.

* * * * *